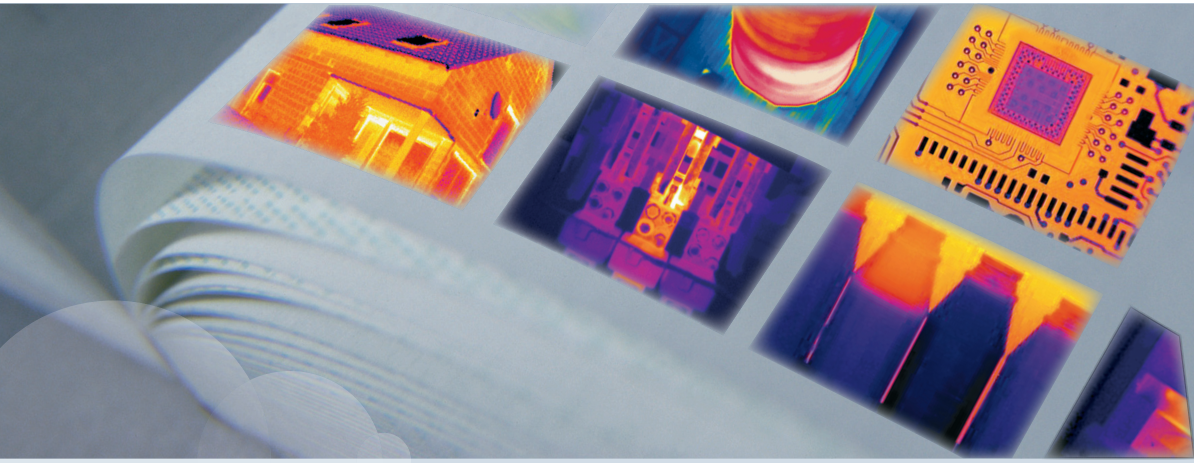
User's Manual



ThermoVision™ LabVIEW® Toolkit

Program version 3.3.7

Publ. No.	T559015
Revision	a590
Language	English (EN)
Issue date	February 29, 2012

ThermoVision™ LabVIEW® Toolkit

User's Manual

Publ. No. T559015 Rev. a590 – ENGLISH (EN) – February 29, 2012

Legal disclaimer

All products manufactured by FLIR Systems are warranted against defective materials and workmanship for a period of one (1) year from the delivery date of the original purchase, provided such products have been under normal storage, use and service, and in accordance with FLIR Systems instruction.

Uncooled handheld infrared cameras manufactured by FLIR Systems are warranted against defective materials and workmanship for a period of two (2) years from the delivery date of the original purchase, provided such products have been under normal storage, use and service, and in accordance with FLIR Systems instruction, and provided that the camera has been registered within 60 days of original purchase.

Detectors for uncooled handheld infrared cameras manufactured by FLIR Systems are warranted against defective materials and workmanship for a period of ten (10) years from the delivery date of the original purchase, provided such products have been under normal storage, use and service, and in accordance with FLIR Systems instruction, and provided that the camera has been registered within 60 days of original purchase.

Products which are not manufactured by FLIR Systems but included in systems delivered by FLIR Systems to the original purchaser, carry the warranty, if any, of the particular supplier only. FLIR Systems has no responsibility whatsoever for such products.

The warranty extends only to the original purchaser and is not transferable. It is not applicable to any product which has been subjected to misuse, neglect, accident or abnormal conditions of operation. Expendable parts are excluded from the warranty.

In the case of a defect in a product covered by this warranty the product must not be further used in order to prevent additional damage. The purchaser shall promptly report any defect to FLIR Systems or this warranty will not apply.

FLIR Systems will, at its option, repair or replace any such defective product free of charge if, upon inspection, it proves to be defective in material or workmanship and provided that it is returned to FLIR Systems within the said one-year period.

FLIR Systems has no other obligation or liability for defects than those set forth above.

No other warranty is expressed or implied. FLIR Systems specifically disclaims the implied warranties of merchantability and fitness for a particular purpose.

FLIR Systems shall not be liable for any direct, indirect, special, incidental or consequential loss or damage, whether based on contract, tort or any other legal theory.

This warranty shall be governed by Swedish law.

Any dispute, controversy or claim arising out of or in connection with this warranty, shall be finally settled by arbitration in accordance with the Rules of the Arbitration Institute of the Stockholm Chamber of Commerce. The place of arbitration shall be Stockholm. The language to be used in the arbitral proceedings shall be English.

Copyright

Quality assurance

The Quality Management System under which these products are developed and manufactured has been certified in accordance with the ISO 9001 standard.

FLIR Systems is committed to a policy of continuous development; therefore we reserve the right to make changes and improvements on any of the products described in this manual without prior notice.

Table of contents

1 Notice to user

Typographical conventions

This manual uses the following typographical conventions:

- **Semibold** is used for menu names, menu commands and labels, and buttons in dialog boxes.
- *Italic* is used for important information.
- `Monospace` is used for code samples.
- UPPER CASE is used for names on keys and buttons.

User-to-user forums

Exchange ideas, problems, and infrared solutions with fellow thermographers around the world in our user-to-user forums. To go to the forums, visit:

http://www.infraredtraining.com/community/boards/

Additional license information

This software is sold under a single user license. This license permits the user to install and use the software on any compatible computer provided the software is used on only one computer at a time. One (1) back-up copy of the software may also be made for archive purposes.

INTENTIONALLY LEFT BLANK

2 Customer help

General

For customer help, visit:

http://support.flir.com

Submitting a question

To submit a question to the customer help team, you must be a registered user. It only takes a few minutes to register online. If you only want to search the knowledgebase for existing questions and answers, you do not need to be a registered user.

When you want to submit a question, make sure that you have the following information to hand:

- The camera model
- The camera serial number
- The communication protocol, or method, between the camera and your PC (for example, HDMI, Ethernet, USB™, or FireWire™)
- Operating system on your PC
- Microsoft® Office version
- Full name, publication number, and revision number of the manual

Downloads

On the customer help site you can also download the following:

- Firmware updates for your infrared camera
- Program updates for your PC software
- User documentation
- Application stories
- Technical publications

INTENTIONALLY LEFT BLANK

3 Overview

3.1 Means of communications

- Firewire 16-bit
- Firewire 8-bit
- Ethernet 16-bit
- Ethernet 8-bit
- iPort/GEV
- USB video

3.2 Supported cameras

- ThermaCAM SC2000/SC3000
- ThermoVision 160/320
- ThermaCAM SC1000
- ThermaCAM S40/S45/S60/S65
- ThermoVision A20/A40
- Cumulus SC4000/SC6000
- ThermaCAM SC640/SC660
- FLIR A320/A300/A310
- FLIR A325 (A320G)/A315/SC305/SC325
- Indigo Merlin
- Indigo Phoenix
- Indigo Omega
- ThermoVision 1000
- ThermoVision 900
- FLIR GF320/GF309
- FLIR T series
- FLIR A615/A645/SC645/SC655
- FLIR Exx series
- FLIR T620/T640

If in doubt, please contact FLIR Customer Support at http://support.flir.com. Contact information can be found at *http://www.flir.com*.

3.3 Main features

- Supports communication and broadcasting via FireWire™, Ethernet, and USB
- Gives the user full control of the camera
- Allows the user to set alarm conditions and measurement functions in the camera
- Allows the user to define I/O functionality (FLIR A3X0, A3X5, A615, and SC6X5)
- Based on ActiveX technology

- Supports acquisition of images through FireWire™, Ethernet, and USB interfaces
- Reads from and writes to files in FLIR Systems' proprietary file format and writes to files in FLIR Systems' open floating point format (*.fpf)
- Converts 16-bit raw pixels into temperature pixels for maximum user flexibility
- Allows 16-bit temperature linear outputs from cameras
- Includes method that allows using individual emissivity value correction on any single pixel or combined measuring value – *e.g.* average, minimum *etc*.
- Supports conditional recording to file with FireWire™, Ethernet, and USB interfaces

3.4 True Temperature Analysis

The ThermoVision™ LabVIEW® Toolkit is a set of VIs (virtual instruments) related to cameras supporting alarms, measurement function and I/O functionality. As you develop in LabVIEW®, you can use these VIs as sub-VIs to manage the communications with a FLIR Systems IR camera in digital mode. You can also generate true temperature images from images acquired through LabVIEW®, so you can use the LabVIEW® IR Measurement and Display tools to analyze the temperatures of the imaged objects.

The ThermoVision™ LabVIEW® Toolkit provides the functions needed to:

- Set up communications between your LabVIEW® VI and the FLIR Systems IR camera
- Capture and gather images via FireWire™ or Ethernet interfaces
- Adjust the camera configuration parameters and focus as you view a live image
- Control the camera calibration
- Send any other camera command to the camera
- Generate a true temperature image from a 16-bit image acquired from the frame-grabber, or using FireWire™, Ethernet, and USB interfaces
- Close the communications to the IR camera

3.5 System requirements

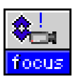
ThermoVision™ LabVIEW® Toolkit requires:

- Windows® XP, 32- or 64-bit, SP2
- Windows® Vista, 32- or 64-bit
- Windows® 7, 32- or 64-bit
- An installed and registered version of National Instruments LabVIEW® 7.1 or later, NI-IMAQ 3.1.3 or later, and IMAQ Vision 7.1 or later
- An installed and registered version of ThermoVision™ LabVIEW® Toolkit.
- A FireWire™, Ethernet, or USB interface
- A FLIR Systems IR camera connected to an Ethernet, FireWire™, or USB port on the computer

4 Overview of ThermoVision™ LabVIEW® Toolkit VIs

For more information about these VIs, refer to section 6 – Description of VIs on page 17.

4.1 General VIs

Open		Opens and establishes a connection to the FLIR Systems IR camera.
Close		Disconnects communication with the camera.
GetVersion		Returns the Camera Control and ThermoVision™ program versions.
GetError		Converts a ThermoVision™ error code to a formatted error string.
GetCameraEvent		Returns the camera events.
GetActiveXReference		Returns the CamCtrl.ocx reference (ActiveX).
SetFocus		Controls the focus state.
GetFocus		Returns the focus absolute position (depends on camera type).
CameraAction		Performs a camera action.
GetCameraParameters		Returns camera parameters.
SetCameraParameters		Configures the camera parameters.
GetDisplayParameters		Reads the display parameters.

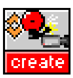
SetDisplayParameters	display	Configures the display parameters.
GetObjectParameters	object	Returns the display parameters.
SetObjectParameters	object	Configures the display parameters.
GetCalibrationParameters	calibr.	Returns the calibration parameters.
SetCalibrationParameters	calibr.	Configures the calibration parameters.
SetResourceValue	ResVal	Sets resource values on some cameras. **NOTE:** Resource protocol not valid for Omega, Cumulus, Phoenix, and Merlin cameras. See the manual for more information.
GetResourceValue	ResVal	Gets resource values on some cameras. **NOTE:** Resource protocol not valid for Omega, Cumulus, Phoenix, and Merlin cameras. See the manual for more information.
LoadIRFile	load IR	Loads an IR image file from file path to the camera control.
GetSequenceFileParameters	seqfile	Gets sequence file parameters.
SetSequenceFileParameters	seqfile	Sets sequence file parameters.
Digital GetLUT	LUT	The table translates raw digital image pixels to temperature. See also section '1.5 Pixel definitions'.
Digital GetAbsLUT	AbsLUT	The table translates raw digital image pixels to absolute pixels. See also section '1.5 Pixel definitions'.
Digital GetObjLUT	ObjLUT	The table translates raw digital image pixels to object pixels. See also section '1.5 Pixel definitions'.
Digital ToTemperature	raw→T	Converts a given raw pixel value to temperature in Kelvin. See also section '1.5 Pixel definitions'.

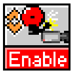
Digital GetImage		Returns an 2D array with pixel values from the camera.
Digital SetImage		Sets an image with raw image pixel values.
DigitalSetImageMode		Configures the camera to send different types of images (A series cameras only).
Digital RecordingAction.		Performs a recording action.
Digital GetRecordingParameters		Returns the recording parameters.
Digital SetRecordingParameters		Configures the recording parameters.
Digital SetImageSize		Pre-allocates image buffers. Used when multiple images should be acquired with Digital GetImages-VI.
Digital GetImageSize		Starts, acquires, and releases a sequence acquisition. Use this VI to capture single or multiple images at a high speed.

4.2 VIs only intended for cameras supporting internal alarms

Applicability:

A20/A300/A310/A320/A40

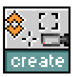
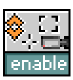
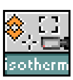
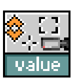
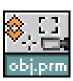
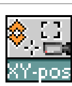
CameraAlarm Create		Creates a camera alarm reference
CameraAlarm Destroy		Destroys a camera alarm reference
CameraAlarm GetAction		Returns the alarm actions
CameraAlarm SetAction		Sets the alarm action
CameraAlarm GetCondition		Returns the alarm conditions

CameraAlarm SetCondition		Sets the alarm condition
CameraAlarm SetEnable		Enables or disables the alarm
CameraAlarm GetStatus		Returns the alarm status
CameraAlarm GetConfiguration.vi		Returns mail and FTP settings
CameraAlarm SetConfiguration.vi		Sets mail and FTP configuration

4.3 *VIs only intended for cameras supporting internal measurement functions*

Applicability:

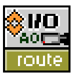
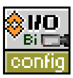
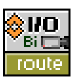
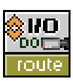
A20/A300/A310/A320/A40/S40/S45/S60/S65/SC640/SC660

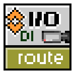
CameraMeasFunc Create		Creates a measurement function reference (spot, box, difference, isotherm, etc.)
CameraMeasFunc Destroy		Destroys a measurement function reference
CameraMeasFunc Difference		Configures the difference measurement
CameraMeasFunc Enable		Enables or disables measurement functions
CameraMeasFunc Isotherm		Configures the isotherm
CameraMeasFunc Measurement		Returns the measurement values
CameraMeasFunc ObjectParameters		Configures object parameters for the measurement function
CameraMeasFunc Position		Positions the measurement function

4.4 VIs only intended for cameras with I/O functions

Applicability:

A20/A300/A310/A315/A320/A320G/A325/A40/A615/SC305/SC325/SC645/SC655

CameraPorts Create		Creates an I/O port reference
CameraPorts Destroy		Destroys the I/O port reference
CameraPorts AnalogIn-putConfig		Configures an analog input channel
CameraPorts AnalogIn-putReadValue		Reads the value from an analog input channel
CameraPorts Analo-gOutputConfig		Configures an analog output channel
CameraPorts Analo-gOutputSignalRoute		Routes a camera signal or function to an analog output channel
CameraPorts Analo-gOutputWriteValue		Writes a value to an analog output channel
CameraPorts Digital-BiDirConfig		Configures a digital bi-directional channel
CameraPorts Digital-BiDirreadvalue		Reads the value from a digital bi-directional channel
CameraPorts Digital-BiDirSignalSource		Routes a camera signal or function to a digital bi-directional channel
CameraPorts Digital-BiDirWriteValue		Writes a value to a bi-directional channel
CameraPorts DigitalIn-putReadValue		Read the value from a digital input channel
CameraPorts DigitalOut-putSignalRoute		Routes a camera signal or function to a digital output channel
CameraPorts DigitalOut-putWriteValue		Writes a value to a digital output channel

CameraPorts DigitalIn-putSignalRoute.vi		Configures digital input actions

4.5 Other VIs

ThermoVision Recti-fyFrameRate.vi		Solves discrepancy between reported and actual frame rates found in some old cameras.

4.6 Pixel definitions

- Raw pixels: Pixels direct from the camera which are not temperature drift-compensated.
- Absolute pixels: Temperature drift-compensated pixels.
- Object pixels: Pixels corrected for emissivity, atmosphere transmission and background radiation.
- Temperature pixels: Pixels in temperature.

5 Examining the example programs

5.1 *Graphical overview of the example programs*

10546803;a4

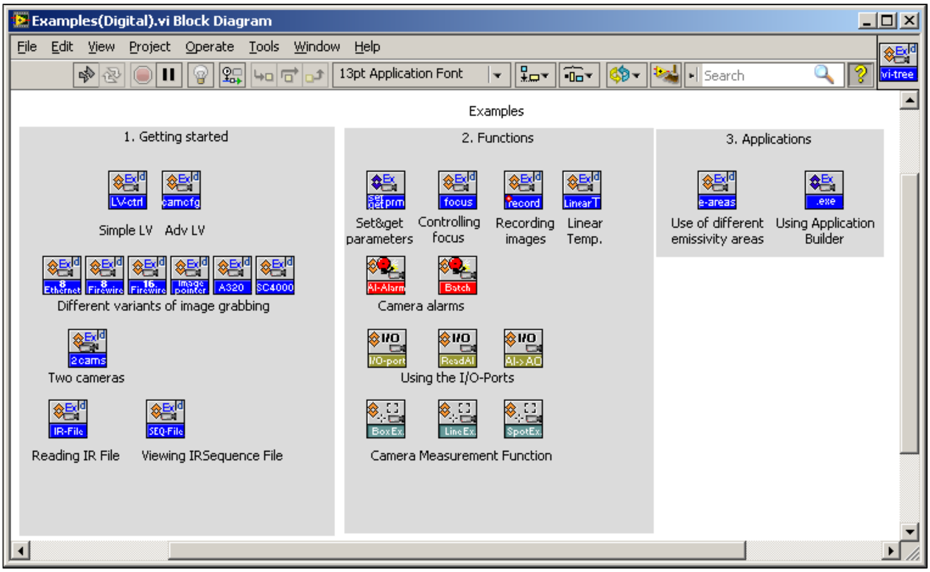


Figure 5.1 Overview

The examples can be found in the following folder:

C:\Program Files\National Instruments\LabVIEW [program version]\examples\ThermoVision

5.2 *Brief description of the 'Getting started' examples*

5.2.1 CameraControl LabVIEWGUI_Simple(Digital).vi

Gets your camera connected to the computer by using the ThermoVision™ LabVIEW® Toolkit VIs.

5.2.2 CameraControl LabVIEWGUI_Advanced(Digital).vi

This example lets you configure some of the camera parameters and take control over the focus. The example is used as a sub-VI in the Emissivity-example.

5.2.3 ImageGrab Ethernet(8 bits image).vi

This example shows how you can grab 8-bits IR-images over Ethernet.

5.2.4 ImageGrab Firewire(8 bits image).vi

This example shows how you can grab 8-bits IR-images over a FireWire™ interface.

5.2.5 ImageGrab Firewire(16 bits image).vi

This example shows how you can grab 16-bits IR-images over a FireWire™ interface.

5.2.6 ImageGrab Firewire(Using image pointer).vi

This example shows how you can perform high speed image acquisitions over a FireWire™ interface by using the ThermoVision GetImages-VI.

5.2.7 ImageGrab A320(Using image pointer).vi

This example shows how you can perform image acquisition with one of most common configurations: TCP/IP - connected A3xx camera.

5.2.8 ImageGrab SC4000(Using image pointer).vi

This example shows how you can perform image acquisition with one of most common configurations: Gigabit Ethernet - connected SC4000/SC6000 camera.

5.2.9 ImageGrab TwoCameras(Digital).vi

This example shows how you can grab images from two cameras.

5.2.10 Read IR File.vi

This example shows how you can read a single-image file.

5.2.11 Read SEQ File.vi

This example shows how you can view a sequence file (.seq).

5.3 Brief description of the ‘Functions’ examples

5.3.1 SetAndGetParameters

This example shows how you can change some of the parameter values in the camera.

5.3.2 Focus

This example shows how you control the camera focus.

5.3.3 Recording

This example gives you a simple overview of the recording possibilities, *e.g.* saving some of the IR images to a file, adding trigger condition etc.

5.3.4 Linear Temperature Image.vi

NOTE: Applicable only for cameras having corresponding build-in functions. See manual for your camera.

This example shows how you can acquire different types of images from the camera.

5.3.5 CameraAlarms AI Alarm Example.vi

NOTE: Applicable only for cameras having corresponding build-in functions. See manual for your camera.

A camera alarm is configured to be activated, if the analog input signal is higher than 2.5 Volt.

5.3.6 CameraAlarms Batch Alarm Example.vi

NOTE: Applicable only for cameras having corresponding build-in functions. See manual for your camera.

This example shows how you can use the batch alarm as a "pre-condition" to normal alarms.

5.3.7 CameraPorts IOPort Configuration Example.vi

NOTE: Applicable only for cameras having corresponding build-in functions. See manual for your camera.

This example shows how you can configure the I/O ports

5.3.8 CameraPorts AI Read Example.vi

NOTE: Applicable only for cameras having corresponding build-in functions. See manual for your camera.

The analog input channel is read and displayed

5.3.9 CameraPorts Connect AI to AO Example.vi

NOTE: Applicable only for cameras having corresponding build-in functions. See manual for your camera.

This example shows how you can route analog input channel 1 to analog output channel 1 on the camera.

5.3.10 CameraMeasFunc Box Example.vi

NOTE: Applicable only for cameras having corresponding build-in functions. See manual for your camera.

This example shows how you use the measurement function **Box**.

5.3.11 CameraMeasFunc Line Example.vi

NOTE: Applicable only for cameras having corresponding build-in functions. See manual for your camera.

This example shows how you use the measurement function **Line**.

5.3.12 CameraMeasFunc Spot Example.vi

NOTE: Applicable only for cameras having corresponding build-in functions. See manual for your camera.

This example shows how you use the measurement function **Spot**.

5.4 Brief description of the 'Application' examples

5.4.1 Emissivity

This example lets you define areas with different emission factors in the IR-image.

5.4.2 Using Application Builder

This example shows how you can build stand alone applications with LabVIEW Application Builder and ThermoVision LabVIEW Analog Toolkit. The included Word document describes how you configure Application Builder and shows which files you must include in your stand alone application.

You need to have LabVIEW Application Builder installed to run this example

6 Description of VIs

6.1 *ThermoVision Open.vi*

Creates and initializes a new ThermoVision object.

This object should be destroyed using a ThermoVision Close VI.

10476103;a1

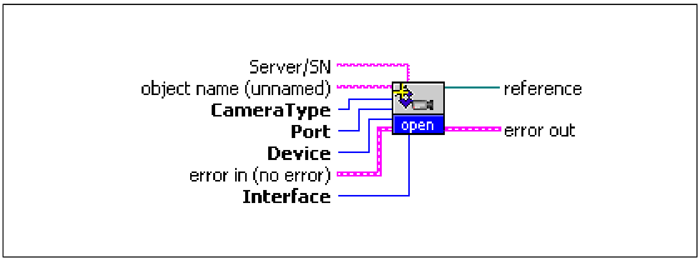


Figure 6.1 Connector Pane

Figure 6.2 Controls and Indicators

	**error in (no error)** error in is a cluster that describes the error status before this VI executes
	**Port** Port to use for connection **0** = Automatic Detection (only valid for device FLIR PC-Card™ and Ethernet) **1..256** = Use COM1-COM256 1024... = TCP/IP port number(or use **0** for automatic detection) **-1** = No connection attempted. Use camera defaults
	**Device** The device type 0= No device (1= FLIR PCCard frame grabber): No longer supported (2= ITEX IC-DIG 16 frame grabber): No longer supported 3= Firewire 16-bit images 4= Firewire 8-bit images 5= Ethernet 16-bit images(Not valid for A20 or A40) 6= Ethernet 8-bit images (7= IR-FlashLink): No longer supported 8 = iPort/GEV 9 = USB video

	CameraType **1** ThermaCAM SC2000/SC3000 **2** ThermoVision 160/320 **3** ThermaCAM SC1000 **4** ThermaCAM S40/S45/S60/S65 **5** ThermoVision A20/A40 **6** Cumulus SC4000/SC6000 **7** ThermaCAM SC640/SC660 **8** FLIR A320/A300/A310 **9** FLIR A325 (A320G)/A315/SC305/SC325 **10** Indigo Merlin **11** Indigo Phoenix **12** Indigo Omega **13** ThermoVision 1000 **14** ThermoVision 900 **15** FLIR GF320/GF309 **16** FLIR T series **17** FLIR A615/A645/SC645/SC655 **18** FLIR Exx series **19** FLIR T620/T640
	Server/SN **Ethernet-connection:** Type the camera server name or IP-address. **FireWire™ and more than one camera on the bus:** Type the serial number of the camera you want to communicate with, in Server/SN-control.
	object name (unnamed) The name of the object to be created
	Interface The Communication Interface Type: 0. File Only. No camera communication, stored images can be read from disk. (1. Serial RS232 camera communication) : No longer supported 2. TCP/IP (Ethernet) 3. AV/C(FireWire) 4. GigaBit Ethernet 5. AXIS Video Server 2401 (Ethernet) 6. UVC (USB video class)
	error out error out is a cluster that describes the error status after this VI executes.
	reference

6.2 ThermoVision Close.vi

Destroys a ThermoVision created with ThermoVision Open VI.

10476903;a1

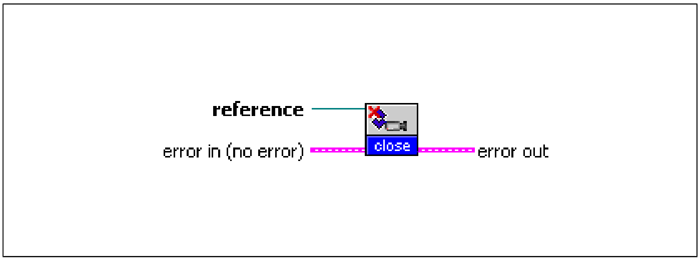
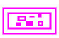


Figure 6.3 Connector Pane

Figure 6.4 Controls and Indicators

	**error in (no error)** error in is a cluster that describes the error status before this VI executes.
	**reference**
	**error out** error out is a cluster that describes the error status after this VI executes.

6.3 ThermoVision GetVersion.vi

Returns Camera Control(ActiveX), ThermoVision and LabVIEW version.

10477003;a1

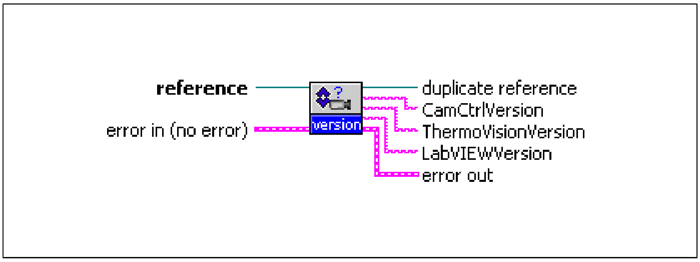


Figure 6.5 Connector Pane

Figure 6.6 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
	error out error out is a cluster that describes the error status after this VI executes.
	duplicate reference
	CamCtrlVersion
	ThermoVisionVersion
	LabVIEWVersion

6.4 ThermoVision GetError.vi

Converts the error code to a formatted error string.

10477203;a1

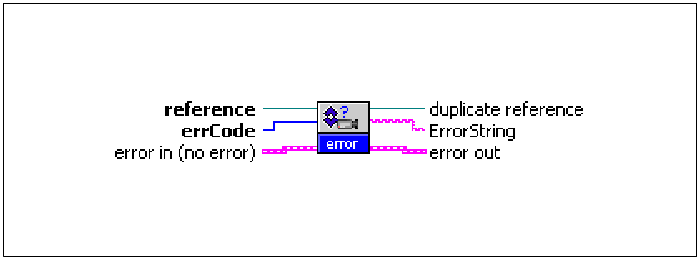


Figure 6.7 Connector Pane

Figure 6.8 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
I16	**errCode** The error code
	error out error out is a cluster that describes the error status after this VI executes
	duplicate reference
abc	**ErrorString** The formatted error string

6.5 ThermoVision GetCamCmdReplyEvent.vi

The CamCmdReply event occurs when the camera control receives a response from a user command issued from the SendCameraCommand-vi.

10477303;a1

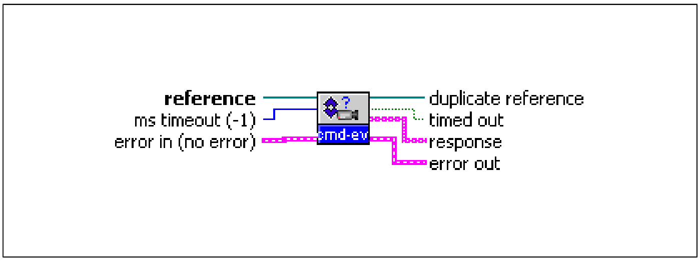


Figure 6.9 Connector Pane

Figure 6.10 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
I32	**ms timeout (-1)** indicates how many milliseconds to wait on Event Queue for an event to arrive.
	error out error out is a cluster that describes the error status after this VI executes.
	duplicate reference
[abc]	**response** response from call to method SendCameraCommand
TF	**timed out** timed out indicates whether the event timed out.

6.6 ThermoVision GetCameraEvent.vi

Returns the camera events:

10477703;a1

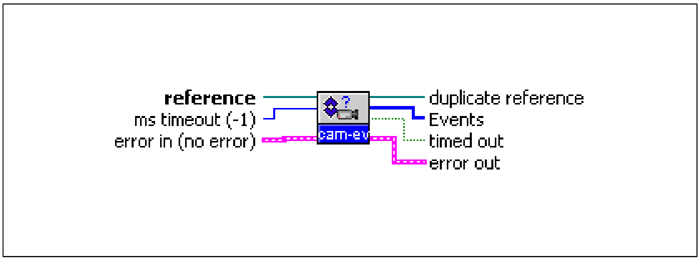


Figure 6.11 Connector Pane

Figure 6.12 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
I32	**ms timeout (-1) ms timeout** indicates how many milliseconds to wait on Event Queue for an event to arrive.
	error out error out is a cluster that describes the error status after this VI executes
	duplicate reference

	Events 0. Not Used 1. Not Used 2. CONNECTED 3. DISCONNECTED 4. CONNECTION_BROKEN 5. RECONNECTED 6. DISCONNECTING 7. AUTOADJUST 8. RECALIB_START 9. REACLIB_STOP 10. LUT_UPDATED 11. REC_UPDATED 12. IMAGE_CAPTURED 13. INIT_COMPLETED 14. FRAME_RATE_TBL_AVAIL 15. FRAME_RATE_CHANGED 16. MEAS_RANGE_TBL_AVAIL 17. MEAS_RANGE_CHANGED 18. IMAGE_SIZE_CHANGED
	Event
	timed out indicates that no event has been received

6

6.7 ThermoVision GetActiveXReference.vi

Returns the CamCtrl.ocx reference(ActiveX).

10485703;a1

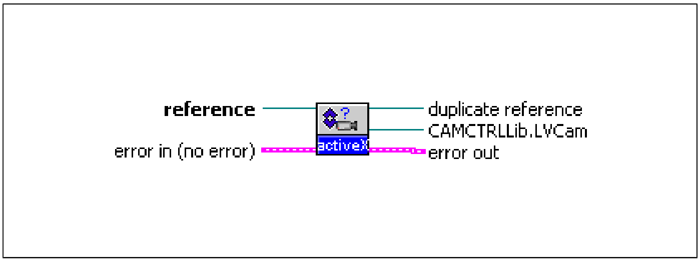


Figure 6.13 Connector Pane

Figure 6.14 Controls and Indicators

	**error in (no error)** error in is a cluster that describes the error status before this VI executes.
	**reference**
	**error out** is a cluster that describes the error status after this VI executes.
	**duplicate reference**
	**CAMCTRLLib.LVCam** The reference to CamCtrl.ocx

6.8 *ThermoVision SetFocus.vi*

Controls the focus mechanism in the camera.

10478103;a1

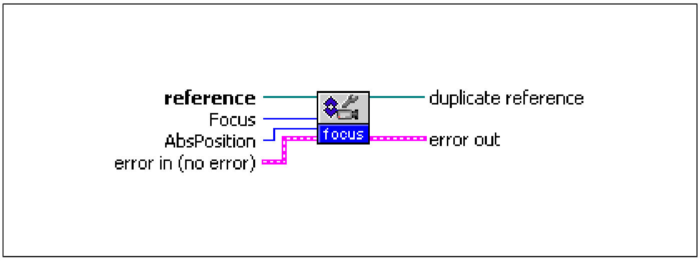


Figure 6.15 Connector Pane

Figure 6.16 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
	Focus Focus State(0=Far, 1=Near,2=Stop, 3=Absolute Position, 4=Auto focus)
	AbsPosition Focus absolute position(depends on camera type)This value is used if Focus-control is set to "Absolute Position".
	error out error out is a cluster that describes the error status after this VI executes.
	duplicate reference

6.9 *ThermoVision GetFocus.vi*

Returns the focus absolute position.

10478203;a1

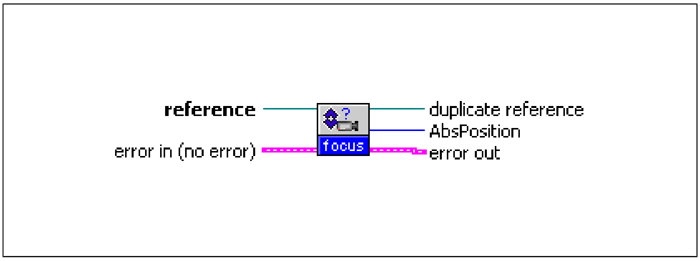


Figure 6.17 Connector Pane

Figure 6.18 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
	error out error out is a cluster that describes the error status after this VI executes.
	duplicate reference
I32	**AbsPosition** Focus absolute position(depends on camera type)

6.10 ThermoVision CameraAction.vi

Performs a specific camera action.

10478403;a1

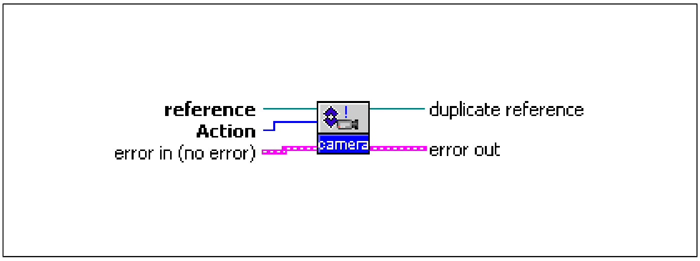


Figure 6.19 Connector Pane

Figure 6.20 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
	Action 0=Internal image correction 1=External image correction 2=Auto adjust 3=Show camera information dialog box 4=Show device status dialog box 5=Reload calibration from camera (SC1000, not supported)
	error out error out is a cluster that describes the error status after this VI executes.
	duplicate reference

6.11 ThermoVision GetCameraParameters.vi

Returns camera parameters.

10478503;a1

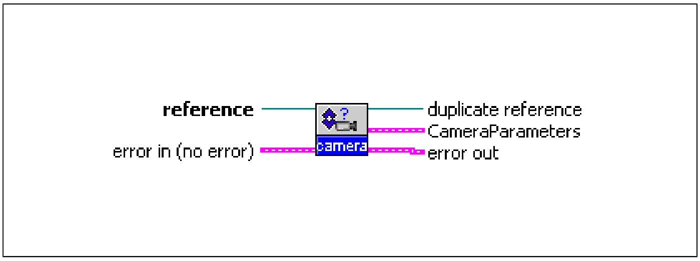


Figure 6.21 Connector Pane

Figure 6.22 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	
	CameraParameters	
	I16	**COM-Port** Port used for connection.(**Read only**) **0** = Automatic Detection **-1** = No connection attempted. Use camera defaults
		DeviceType The device type 0= No device (1= FLIR PCCard frame grabber): No longer supported (2= ITEX IC-DIG 16 frame grabber): No longer supported 3= Firewire 16-bit images 4= Firewire 8-bit images 5= Ethernet 16-bit images(Not valid for A20 or A40) 6= Ethernet 8-bit images (7= IR-FlashLink): No longer supported 8 = iPort/GEV 9 = USB video

6

		CameraType 0 = ThermaCAM SC2000/SC3000 (not supported) 1 = THV 320/160 (not supported) 2 = ThermaCAM SC1000 (not supported) 3 = ThermaCAM S40/S45/S60/S65 4 = Thermovision A20/A40 5 = Cumulus SC4000/SC6000 6 = ThermaCAM SC640/CS660 7 = FLIR A320/A300/A310 8 = FLIR A325(A320G)/A315/SC305/SC325 9 = Indigo Merlin 10 = Indigo Phoenix 11 = Indigo Omega 12 = THV 1000 (not supported) 13 = THV 900 (not supported) 14 = FLIR GF320/GF309 15 = FLIR T-series 16 = FLIR A615/SC645/SC655
	abc	**ModelName** The camera model name.(**Read only**)
	abc	**VideoMode** Current video mode(**Read only**)
	abc	**LensName** The name of the lens(**Read only**)
	abc	**BatteryStatus** The status of the camera battery.(**Read only**)
	I16	**MeasurementRange** The selected measurement range index, see MeasurementRanges
	[abc]	**MeasurementRanges** List of measurement ranges in Kelvin(Read only)
		CoolerStatus The status of the cooler (0=On, 1=Off, 2=Standby, 3=Cooling)
	abc	**CameraPalette** Current camera palette
	I16	**NoiseReduction** Noise reduction(SC2000) **0-2**=Off **3-5** =Normal >5 =High
	I32	**FrameRate** Frame rate or image speed(in Hertz)

		FieldMode Field Mode(only AGEMA 550) **0** = Normal Motion Targets **1** = Slow Motion Targets
	DBL	**FrameRate** Frame rate or image speed(in Hertz)
	[DBL]	**AvailableFrameRates** List of available frame rates that the camera supports

6.12 *ThermoVision SetCameraParameters.vi*

Configures the camera control parameters.

10478703;a1

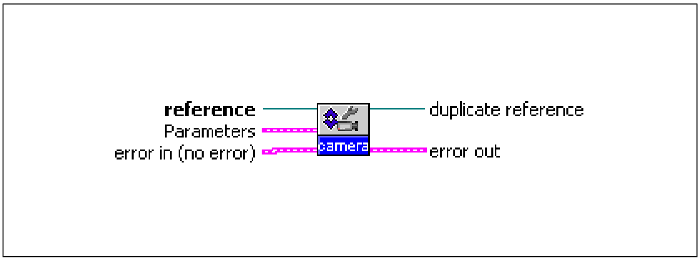


Figure 6.23 Connector Pane

Figure 6.24 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	Parameters	
	I16	**COM-Port** Port used for connection.(**Read only**) **0** = Automatic Detection (only valid for device FLIR PC-Card™ and Ethernet) **1..256** = Use COM1-COM256 **1024...** = TCP/IP port number(or use 0 for automatic detection) **-1** = No connection attempted. Use camera defaults
		DeviceType The device type 0= No device (1= FLIR PCCard frame grabber): No longer supported (2= ITEX IC-DIG 16 frame grabber): No longer supported 3= Firewire 16-bit images 4= Firewire 8-bit images 5= Ethernet 16-bit images(Not valid for A20 or A40) 6= Ethernet 8-bit images (7= IR-FlashLink): No longer supported 8 = iPort/GEV 9 = USB video

		CameraType 0 = ThermaCAM SC2000/SC3000 (not supported) 1 = THV 320/160 (not supported) 2 = ThermaCAM SC1000 (not supported) 3 = ThermaCAM S40/S45/S60/S65 4 = Thermovision A20/A40 5 = Cumulus SC4000/SC6000 6 = ThermaCAM SC640/CS660 7 = FLIR A320/A300/A310 8 = FLIR A325(A320G)/A315/SC305/SC325 9 = Indigo Merlin 10 = Indigo Phoenix 11 = Indigo Omega 12 = THV 1000 (not supported) 13 = THV 900 (not supported) 14 = FLIR GF320/GF309 15 = FLIR T-series 16 = FLIR A615/SC645/SC655
	abc	**ModelName** The camera model name.(**Read only**)
	abc	**VideoMode** Current video mode(**Read only**)
	abc	**LensName** The name of the lens(**Read only**)
	abc	**BatteryStatus** The status of the camera battery.(**Read only**)
	I16	**MeasurementRange** The selected measurement range index, see MeasurementRanges
	[abc]	**MeasurementRanges** List of measurement ranges in Kelvin(**Read only**)
		CoolerStatus The status of the cooler (0=On, 1=Off, 2=Standby, 3=Cooling)
	abc	**CameraPalette** Current camera palette
	I16	**NoiseReduction** Noise reduction(SC2000) **0-2**=Off **3-5** =Normal **>5** =High
	I32	**FrameRate** Frame rate or image speed(in Hertz)

		FieldMode Field Mode(only AGEMA 550) **0** = Normal Motion Targets **1** = Slow Motion Targets
		FrameRate Frame rate or image speed(in Hertz)
		AvailableFrameRates List of available fame rates that the camera supports
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	

6.13 ThermoVision GetDisplayParameters.vi

Returns the display parameters.

10478903;a1

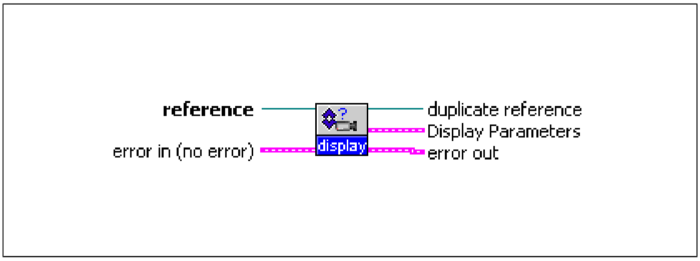


Figure 6.25 Connector Pane

Figure 6.26 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	
	Display Parameters	
	SGL	**HiScale** The high scale limit(in Kelvin). Range 0-5000 K (only on analog video or HDMI output)
	SGL	**LoScale** The low scale limit(in Kelvin). Range 0-5000 K (only on analog video or HDMI output)
	SGL	**Zoom** The zoom factor. Range 1.0 - 8.0
		Scale The Scale Visibility (only on analog video output) **0** = Not Visible **1** = Visible
	abc	**IR-SourceFile** The IR Source File (absolute path).

6

6.14 ThermoVision SetDisplayParameters.vi

Configures the display parameters.

10479103;a1

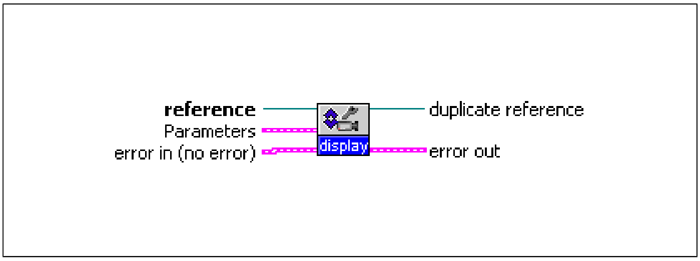


Figure 6.27 Connector Pane

Figure 6.28 Controls and Indicators

		error in (no error) error in is a cluster that describes the error status before this VI executes.
		reference
		Display Parameters
		HiScale The high scale limit(in Kelvin). Range 0-5000 K (only on analog video or HDMI output)
		LoScale The low scale limit(in Kelvin). Range 0-5000 K (only on analog video or HDMI output)
		Zoom The zoom factor. Range 1.0 - 8.0
		Scale The Scale Visibility (only on analog video output) **0** = Not Visible **1** = Visible
		IR-SourceFile The IR Source File (absolute path).
		error out error out is a cluster that describes the error status after this VI executes.
		duplicate reference

6.15 ThermoVision GetObjectParameters.vi

Returns the object parameters.

10479303;a1

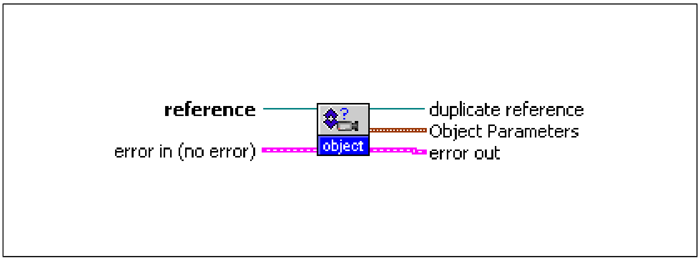


Figure 6.29 Connector Pane

Figure 6.30 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	
	Object Parameters	
	SGL	**AmbTemp** Ambient temperature(in Kelvin). Range 0-5000K
	SGL	**AtmTemp** Atmospheric temperature(in Kelvin). Range 0-5000K
	SGL	**ObjectDist** Object distance(in meter). Range 0-10000m
	SGL	**RelEmissivity** Object emissivity. Range 0.01-1.00
	SGL	**RelHumidity** Relative humidity. Range 0.0-1.0
	SGL	**RefTemp** Reference temperature(in Kelvin). Range 0-5000K
	SGL	**ExtOpticsTemp** External optics temperature(in Kelvin). Range 0-5000K
	SGL	**ExtOpticsTransm** External optics transmission. Range 0.01-1.00

6.16 ThermoVision SetObjectParameters.vi

Configures the object parameters.

10479503;a1

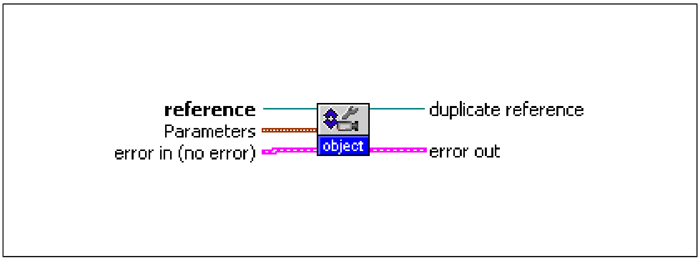


Figure 6.31 Connector Pane

Figure 6.32 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	Parameters	
	SGL	**AmbTemp** Ambient temperature(in Kelvin). Range 0-5000K
	SGL	**AtmTemp** Atmospheric temperature(in Kelvin). Range 0-5000K
	SGL	**ObjectDist** Object distance(in meter). Range 0-10000m
	SGL	**RelEmissivity** Object emissivity. Range 0.01-1.00
	SGL	**RelHumidity** Relative humidity. Range 0.0-1.0
	SGL	**RefTemp** Reference temperature(in Kelvin). Range 0-5000K
	SGL	**ExtOpticsTemp** External optics temperature(in Kelvin). Range 0-5000K
	SGL	**ExtOpticsTransm** External optics transmission. Range 0.01-1.00
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	

6.17 ThermoVision GetCalibrationParameters.vi

Returns the calibration parameters.

NOTE: The automatic temperature compensation is handled in the ActiveX-control, not in the camera. The camera control over the temperature compensation is turned off when the Open-method is executed.

10479703;a1

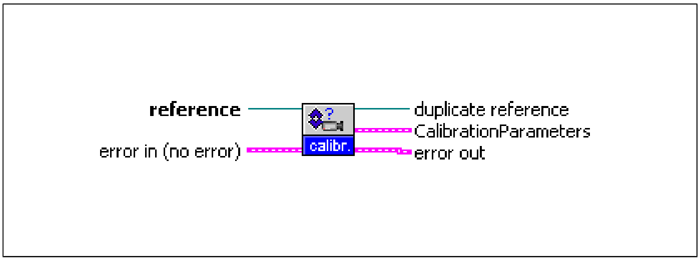


Figure 6.33 Connector Pane

Figure 6.34 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	
	CalibrationParameters	
		Title The calibration title
		AutoTempComp Automatic temperature compensation (0=Off, 1=On) Note! The automatic temperature compensation is handled in the ActiveX-control, not in the camera. The camera control over the temperature compensation is turned off when the Open-method is executed.

6.18 ThermoVision SetCalibrationParameters.vi

Configures the calibration parameters.

NOTE: The automatic temperature compensation is handled in the ActiveX-control, not in the camera. The camera control over the temperature compensation is shut off when connected.

10479803;a1

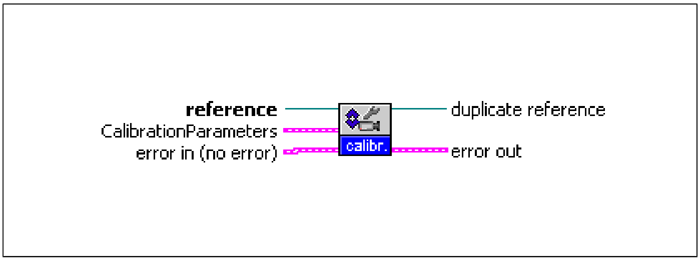


Figure 6.35 Connector Pane

Figure 6.36 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	CalibrationParameters	
		Title The calibration title
		AutoTempComp Automatic temperature compensation (0=Off, 1=On) Note! The automatic temperature compensation is handled in the ActiveX-control, not in the camera. The camera control over the temperature compensation is turned off when the Open-method is executed.
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	

6.19 ThermoVision GetResourceValue.vi

Returns a resource value.

GetResourceValue are valid for the following cameras:

- ThermaCAM S40/S45/S60/S65
- Thermovision A20/A40
- ThermaCAM SC640/CS660
- FLIR A320/A300/A310
- FLIR A325/A320G/A315/SC305/SC325
- FLIR GF320/GF309
- FLIR T-series
- FLIR A615/SC645/SC655

Input parameter:

Resource: The resource path

SEE ALSO: For more information, see the following:

- SXX & AXX Camera Commands manual (Publ. No. 1 557 845)
- ICD A320 Camera-PC manual (Publ. No. T559002)

Output parameter:

Value: The resource value as an ActiveX variant

10479903;a1

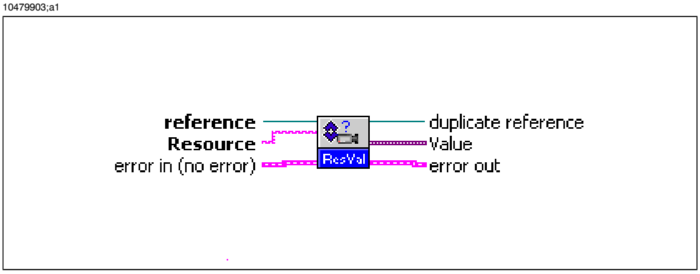


Figure 6.37 Connector Pane

Figure 6.38 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
	Resource
	error out error out is a cluster that describes the error status after this VI executes.
	duplicate reference

	Value

6.20 ThermoVision SetResourceValue.vi

Sets a resource value.

GetResourceValue are valid for the following cameras:

- ThermaCAM S40/S45/S60/S65
- Thermovision A20/A40
- ThermaCAM SC640/CS660
- FLIR A320/A300/A310
- FLIR A325/A320G/A315/SC305/SC325
- FLIR GF320/GF309
- FLIR T-series
- FLIR A615/SC645/SC655

SEE ALSO: For more information, see the following:

- SXX & AXX Camera Commands manual (Publ. No. 1 557 845)
- ICD A320 Camera-PC manual (Publ. No. T559002)

10480103;a1

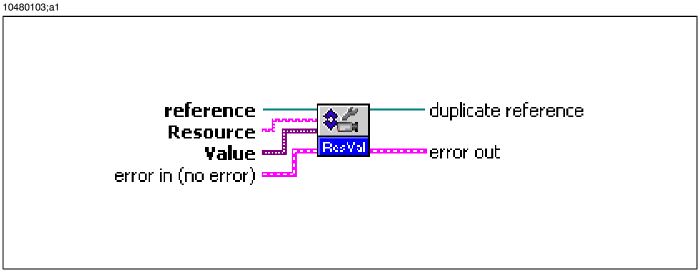


Figure 6.39 Connector Pane

Figure 6.40 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
	Resource
	Value
	error out error out is a cluster that describes the error status after this VI executes.
	duplicate reference

6.21 ThermoVision LoadIRFile.vi

Loads an IR image file from file path to the camera control. Device and Interface parameters in the call to "ThermoVision Open.vi" must be set to "No Device" and "File Only" Otherwise the file is overwritten by the images from the camera.

The parameters to be set:

File Path (absolute)

10480203;a1

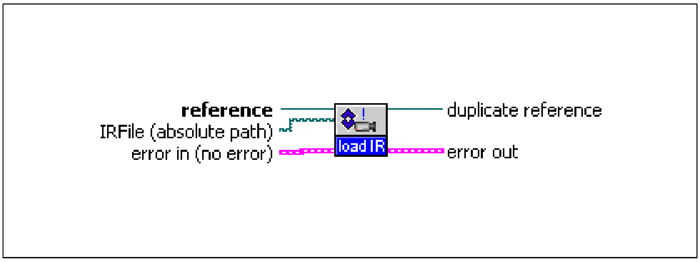


Figure 6.41 Connector Pane

Figure 6.42 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes. If error in indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its error out cluster. If no error has occurred, then this VI executes normally and sets its own error status in error out. Use the error handler VIs to look up the error code and to display the corresponding error message. Using error in and error out clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.	
	TF	**status** status is TRUE if an error occurred before this VI was called, or FALSE if not. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code.
	I32	**code** code is the number identifying an error or warning. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code. Use the error handler VIs to look up the meaning of this code and to display the corresponding error message.
	abc	**source** source is a string that indicates the origin of the error, if any. Usually source is the name of the VI in which the error occurred.
	reference	
	IRFile (absolute path) Empty path will pop up file dialog	

	error out error out is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, error out is the same as error in. Otherwise, error out shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using error in and error out clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.	
	TF	**status** status is TRUE if an error occurred, or FALSE if not. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code.
	I32	**code** code is the number identifying an error or warning. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code. Use the error handler VIs to look up the meaning of this code and to display the corresponding error message.
	abc	**source** source is a string that indicates the origin of the error, if any. Usually source is the name of the VI in which the error occurred.
	duplicate reference	

6.22 ThermoVision GetSequenceFileParameters.vi

Gets sequence file parameters.

The parameters to get:

- Number of images (Read Only)
- Current image number in image sequence file
- Wrapping. True=wrap to the first image in sequence. False=Do not wrap at end of sequence

10480603;a1

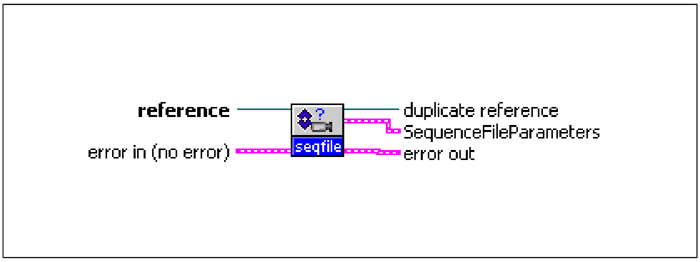


Figure 6.43 Connector Pane

Figure 6.44 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes. If error in indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its error out cluster. If no error has occurred, then this VI executes normally and sets its own error status in error out. Use the error handler VIs to look up the error code and to display the corresponding error message. Using error in and error out clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.	
	TF	**status** status is TRUE if an error occurred before this VI was called, or FALSE if not. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code.
	I32	**code** code is the number identifying an error or warning. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code. Use the error handler VIs to look up the meaning of this code and to display the corresponding error message.
	abc	**source** source is a string that indicates the origin of the error, if any. Usually source is the name of the VI in which the error occurred.
	reference	

	SequenceFileParameters Local Sequence file parameters: Number of images (ReadOnly) Current image number in image sequence file Wrapping. True=wrap to the first image in sequence. False=Do not wrap at end of sequence	
	I32	**NumberOfImages**
	I32	**CurrentImageNumber**
	TF	**Wrapping**
	error out error out is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, error out is the same as error in. Otherwise, error out shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using error in and error out clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.	
	TF	**status** status is TRUE if an error occurred, or FALSE if not. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code.
	I32	**code** code is the number identifying an error or warning. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code. Use the error handler VIs to look up the meaning of this code and to display the corresponding error message.
	abc	**source** source is a string that indicates the origin of the error, if any. Usually source is the name of the VI in which the error occurred.
	duplicate reference	
	SequenceFileParameters Sequence file parameters: Number of images (ReadOnly) Current image number in image sequence file Wrapping. True=wrap to the first image in sequence. False=Do not wrap at end of sequence	
	I32	**NumberOfImages**
	I32	**CurrentImageNumber**
	TF	**Wrapping**

6

6.23 ThermoVision SetSequenceFileParameters.vi

Sets sequence file parameters.

The parameters to be set:

- Number of images (Read Only)
- Current image number in image sequence file
- Wrapping. True=wrap to the first image in sequence. False=Do not wrap at end of sequence

10480703;a1

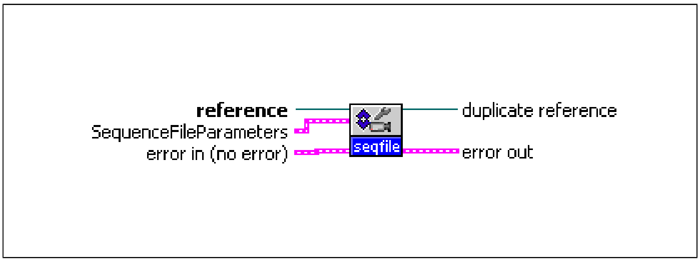


Figure 6.45 Connector Pane

Figure 6.46 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes. If error in indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its error out cluster. If no error has occurred, then this VI executes normally and sets its own error status in error out. Use the error handler VIs to look up the error code and to display the corresponding error message. Using error in and error out clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.	
	TF	**status** status is TRUE if an error occurred before this VI was called, or FALSE if not. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code.
	I32	**code** code is the number identifying an error or warning. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code. Use the error handler VIs to look up the meaning of this code and to display the corresponding error message.
	abc	**source** source is a string that indicates the origin of the error, if any. Usually source is the name of the VI in which the error occurred.
	reference	

[cluster]	**SequenceFileParameters** Sequence file parameters: Number of images (ReadOnly) Current image number in image sequence file Wrapping. True=wrap to the first image in sequence. False=Do not wrap at end of sequence	
	I32	**NumberOfImages**
	I32	**CurrentImageNumber**
	TF	**Wrapping**
[cluster]	**error out** error out is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, error out is the same as error in. Otherwise, error out shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using error in and error out clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.	
	TF	**status** status is TRUE if an error occurred, or FALSE if not. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code.
	I32	**code** code is the number identifying an error or warning. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code. Use the error handler VIs to look up the meaning of this code and to display the corresponding error message.
	abc	**source** source is a string that indicates the origin of the error, if any. Usually source is the name of the VI in which the error occurred.
[refnum]	**duplicate reference**	

6

6.24 ThermoVision Digital GetLUT.vi

Gets a temperature translation table. The table can be used to translate raw image pixels to temperature. The table size depends on the LUT type parameter.

NOTE: To be used with non-FLIR framegrabbers, *e.g.* NI-1422, or similar.

10480803;a1

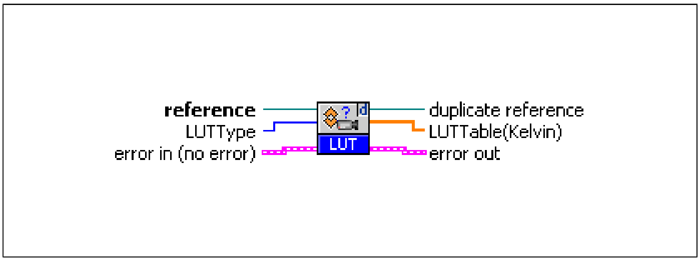


Figure 6.47 Connector Pane

Figure 6.48 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
	LUTType The LUT table size 0=Reserved 1=Table size 65536(16-bit pixels) 2=Table size 32768(15-bit pixels)
	error out error out is a cluster that describes the error status after this VI executes
	duplicate reference
	LUTTable(Kelvin) A 1-dimensional array of single precision floats translating raw image pixels to temperature(in Kelvin)

6.25 ThermoVision Digital GetAbsLUT.vi

Gets a pixel translation table. The table translates raw image pixels to absolute pixels.

NOTE: To be used with non-FLIR framegrabbers, *e.g.* NI-1422, or similar.

10481003;a1

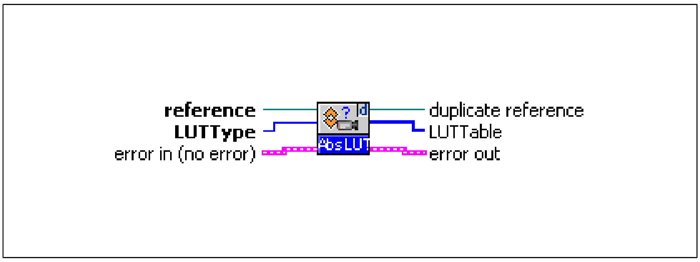


Figure 6.49 Connector Pane

Figure 6.50 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
	LUTType The LUT table size: **0** = Not Used **1** = 65536 (16-bit pixels) **2** = 32768 (15-bit pixels)
	error out error out is a cluster that describes the error status after this VI executes
	duplicate reference
U16	**LUTTable** 1-dimensional array of unsigned integer translating raw image pixels to absolute image pixels.

6.26 ThermoVision Digital GetObjLUT.vi

Gets a pixel translation table. The table translates raw image pixels to object pixels.

NOTE: To be used with non-FLIR framegrabbers, *e.g.* NI-1422, or similar.

10481203;a1

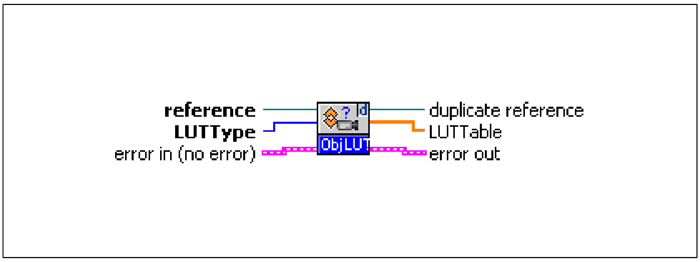


Figure 6.51 Connector Pane

Figure 6.52 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
	LUTType The LUT table size: 0 = Not Used 1 = 65536 (16-bit pixels) 2 = 32768 (15-bit pixels)
	error out error out is a cluster that describes the error status after this VI executes
	duplicate reference
	LUTTable 1-dimensional array of single precision float translating raw image pixels to object image pixels.

6.27 ThermoVision Digital ToTemperature.vi

Converts a given raw pixel value to temperature in Kelvin.

10481303;a1

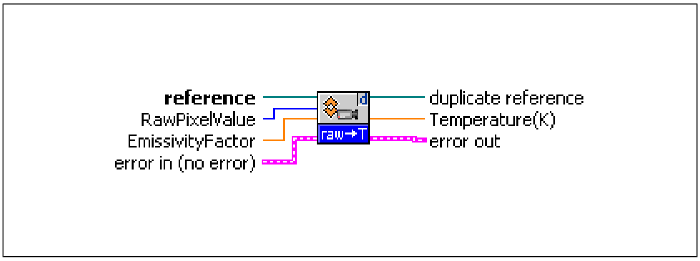


Figure 6.53 Connector Pane

Figure 6.54 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
I32	**RawPixelValue** Raw pixel value
SGL	**EmissivityFactor** Emissivity factor(0.0-1.0). If 0 the image emissivity is used.
	error out error out is a cluster that describes the error status after this VI executes
	duplicate reference
SGL	**Temperature(K)** Temperature in Kelvin

6.28 ThermoVision Digital GetImage.vi

Returns an image with only pixel data from the camera.

Use this method to get the IR image with FLIR image grabber..

10481403;a1

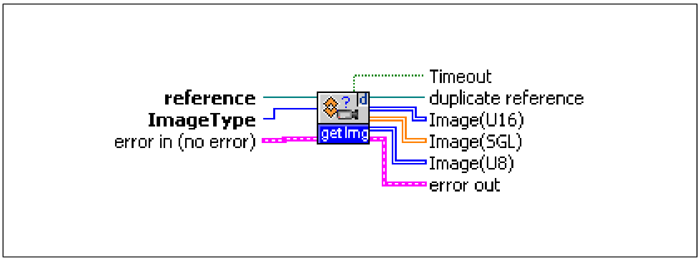


Figure 6.55 Connector Pane

Figure 6.56 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
	ImageType 0=Raw image pixels(16-bit unsigned word) 1=Absolute image pixels(16-bit unsigned word) 2=Object signal pixels(single precision float) 3=Temperature pixels(single precision float)
	error out error out is a cluster that describes the error status after this VI executes
	duplicate reference
[U16]	**Image(U16)** A 2-dimensional array with image pixels. Used when image type is Raw or Absolute.
	U16
[SGL]	**Image(SGL)** A 2-dimensional array with image pixels. Used when image type is Object or Temperature.
[U8]	**Image(U8)** A 2-dimensional array with image pixels. Used when image type is Raw or Absolute.
TF	**Timeout**

6.29 *ThermoVision Digital SetImage.vi*

Sets an image with raw image pixel data.

10481703;a1

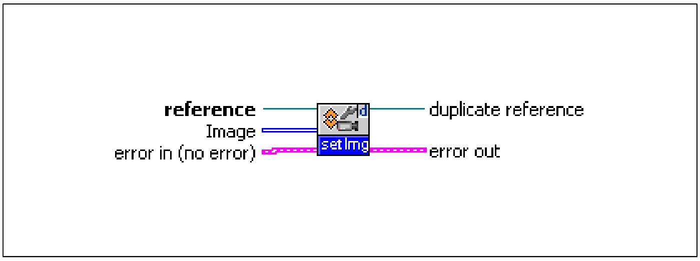


Figure 6.57 Connector Pane

Figure 6.58 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
[U16]	**Image** The raw image in an array of U16.
	U16
	error out error out is a cluster that describes the error status after this VI executes
	duplicate reference

6.30 ThermoVision Digital SetImageMode.vi

Configures the camera to send different types of images (only on A-series).

NOTE: Affects ThermoVision Digital GetImage.vi and ThermoVision Digital GetImages.vi

The parameter to be set:

Image Mode (0=Signal, 1=Temperature 0.1K and 2=Temperature 0.01K)

10481803;a1

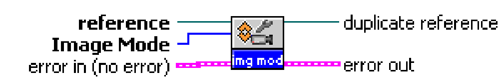
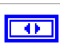


Figure 6.59 Connector Pane

Figure 6.60 Controls and Indicators

	**error in (no error)** error in is a cluster that describes the error status before this VI executes. If error in indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its error out cluster. If no error has occurred, then this VI executes normally and sets its own error status in error out. Use the error handler VIs to look up the error code and to display the corresponding error message. Using error in and error out clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.	
		**status** status is TRUE if an error occurred before this VI was called, or FALSE if not. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code.
		**code** code is the number identifying an error or warning. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code. Use the error handler VIs to look up the meaning of this code and to display the corresponding error message.
		**source** source is a string that indicates the origin of the error, if any. Usually source is the name of the VI in which the error occurred.
	**reference**	
	**Image Mode**	

	error out error out is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, error out is the same as error in. Otherwise, error out shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using error in and error out clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.	
	TF	**status** status is TRUE if an error occurred, or FALSE if not. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code.
	I32	**code** code is the number identifying an error or warning. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code. Use the error handler VIs to look up the meaning of this code and to display the corresponding error message.
	abc	**source** source is a string that indicates the origin of the error, if any. Usually source is the name of the VI in which the error occurred.
	duplicate reference	

6.31 ThermoVision Digital RecordingAction.vi

Performs a recording action.

10481903;a1

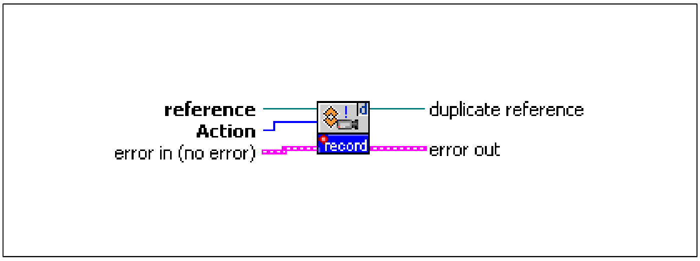


Figure 6.61 Connector Pane

Figure 6.62 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
	Action 0=Start recording 1=Stop recording 2=Enable recording 3=Disable recording 4=Pause recording(not implemented) 5=Resume paused recording(not implemented) 6=Single snapshot recording 7=Show recording settings dialog box 8=Clear recording file list
	error out error out is a cluster that describes the error status after this VI executes.
	duplicate reference

6.32 ThermoVision Digital GetRecordingParameters.vi

Returns the recording parameters.

10482003;a1

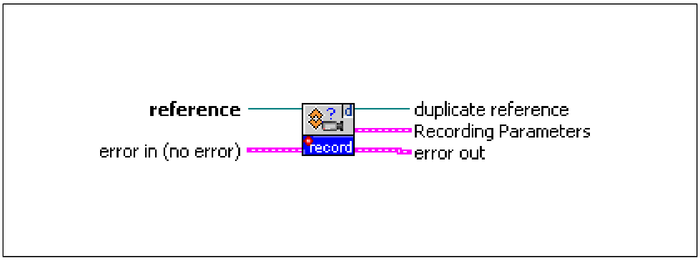


Figure 6.63 Connector Pane

Figure 6.64 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	
	Recording Parameters	
		State Read only parameter 0 = Disabled 1 = Waiting 2 = Active 3 = Paused
		StoreCond Recording store condition(0=User, 1=Highest, 2=Every N:th Image, 3= At External Trigger)
		StartCond Recording start condition(0=User, 1=At Absolute Time, 2= At External Trigger)
		StopCond Recording stop condition(0=User, 1=After Time Interval, 2= After N Images, 3=At External Trigger)
		TrigSource Recording trigger source (0=External device, 1=COM port, 2=LPTport).LPT is not supported in Windows NT
	I16	**TrigPort** Recording Trigger Port. Range 1-256

		FileFormat 0 = Multiple propriety image files 1 = Sequence file format 2 = Multiple public image files
	DBL	**StartValue** Recording Start Value. Delay in seconds if start condition =2(At External Trigger). Absolute time in seconds from 12AM January 1904, if start condition =1(At absolute time)
	DBL	**StoreValue** Recording Store Value. Image interval if store condition = 2(Every N:th image). Time interval in seconds if store condition 3(At Time Interval)
	DBL	**StopValue** Recording Stop Value. Time interval in seconds if stop condition =1(After time interval). Delay in seconds if stop condition =3(At external trig)
	abc	**DirectoryPath** Recording directory path. E.g. "C:\images"
	abc	**FileBaseName** Recording file base name.
	[abc]	**Files** The recorded files(read only parameter)
		PresentationMode Presentation mode during active recording (0=Get Image Disabled, 1=Get Image Enabled)

6.33 ThermoVision Digital SetRecordingParameters.vi

Configures the recording parameters.

10482203;a1

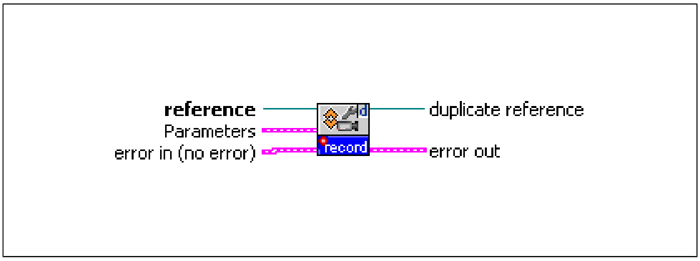


Figure 6.65 Connector Pane

Figure 6.66 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	Parameters	
		State Read only parameter **0** = Disabled **1** = Waiting **2** = Active **3** = Paused
		StoreCond Recording store condition(0=User, 1=Highest, 2=Every N:th Image, 3= At External Trigger)
		StartCond Recording start condition(0=User, 1=At Absolute Time, 2= At External Trigger)
		StopCond Recording stop condition(0=User, 1=After Time Interval, 2= After N Images, 3=At External Trigger)
		TrigSource Recording trigger source (0=Parallel Interface, 1=COM port, 2=LPTport).LPT is not supported in Windows NT
	I16	**TrigPort** Recording Trigger Port. Range 1-256
		FileFormat Recording File Format(0=IMG, 1=FLIR Public Format)
	DBL	**StartValue** Recording Start Value. Delay in seconds if start condition =2(At External Trigger). Absolute time in seconds from 12AM January 1904, if start condition =1(At absolute time)

	DBL	**StoreValue** Recording Store Value. Image interval if store condition = 2(Every N:th image). Time interval in seconds if store condition 3(At Time Interval)
	DBL	**StopValue** Recording Stop Value. Time interval in seconds if stop condition =1(After time interval). Delay in seconds if stop condition =3(At external trig)
	abc	**DirectoryPath** Recording directory path. E.g. "C:\images"
	abc	**FileBaseName** Recording file base name.
	[abc]	**Files** The recorded files(read only parameter)
		abc **String**
		PresentationMode Presentation mode during active recording (0=Get Image Disabled, 1=Get Image Enabled)
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	

6.34 ThermoVision Digital SetImageSize.vi

Pre-allocates image buffers. Used when multiple images should be acquired with "ThermoVision Digital GetImages"-VI.

NOTE: The horizontal resolution of acquired image must be an even multiple of 8 bytes. Normally, this is not a problem if you are acquiring images from an IR-camera(who often returns images in 8 bytes multiple). The problem could occur if you are reading IR images from disk.

10482403;a1

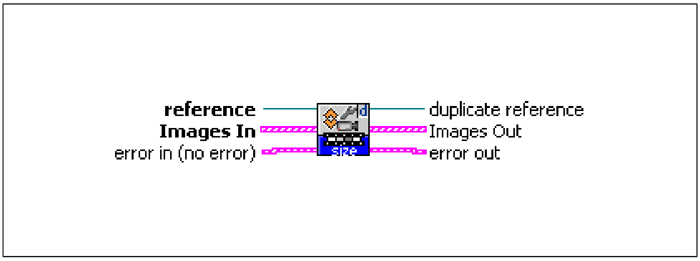


Figure 6.67 Connector Pane

Figure 6.68 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	Images In	
		Image
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	
	Images Out	
		Image

6.35 ThermoVision Digital GetImages.vi

Starts, acquires, and releases a sequence acquisition. Use this VI to capture multiple images.

NOTE: Use "ThermoVision Digital SetImageSize.vi" to allocate image buffers.

10482703;a1

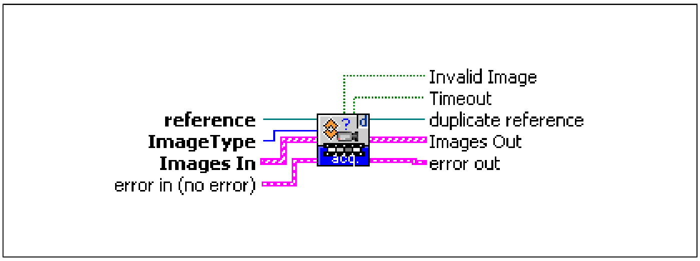


Figure 6.69 Connector Pane

Figure 6.70 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	ImageType 0=Raw image pixels(16-bit unsigned word) 1=Absolute image pixels(16-bit unsigned word) 2=Object signal pixels(single precision float) 3=Temperature pixels(single precision float)	
	Images In	
		Image
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	
	Timeout	
	Images Out	
		Image
	Invalid Image	

6.36 CameraAlarm Create.vi

Creates a new Camera Alarm object.

Used for setting the alarm features in following camera type(s):

- FLIR A20/A40
- FLIR A320/A300/A310

10482803;a1

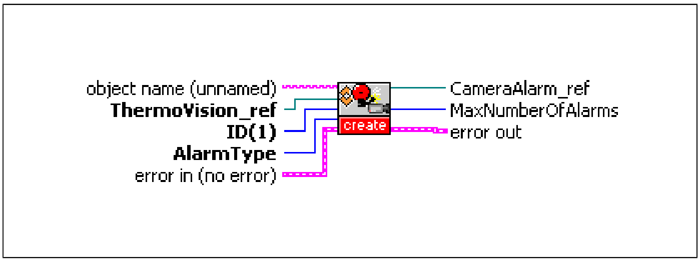


Figure 6.71 Connector Pane

Figure 6.72 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	ThermoVision_ref
	object name (unnamed)
	ID(1) Camera alarm id-number(1,2.. n , n depends on how many alarms your camera can handle). The ID-number have to be unique within the Alarm Type("Normal Alarm" or "Batch Alarm".)
	AlarmType Alarm can be one of following two types. - Normal Alarm: Works as a normal alarm when batch alarms are disabled or when batch alarm conditions are fulfilled. Can be routed to an output pin by using "CameraPorts DigitalOutputRouteSignal"-VI- Batch Alarm(or pre-condition alarm): Use this alarm type if you want to set up the batch alarms. Batch alarms works as a "pre-condition" for the normal alarms. Batch alarms can't be routed to output pins.
	error out error out is a cluster that describes the error status after this VI executes.
	CameraAlarm_ref
	MaxNumberOfAlarms Maximum allowed alarms(of selected Alarm Type) in the camera.

6.37 *CameraAlarm Destroy.vi*

Destroys a CameraAlarm object created with a CameraAlarm constructor VI.

10482903;a1

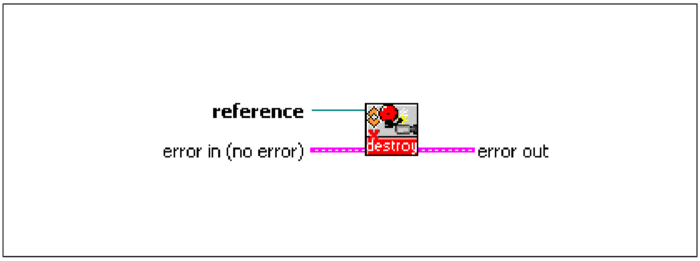


Figure 6.73 Connector Pane

Figure 6.74 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
	error out error out is a cluster that describes the error status after this VI executes.

6.38 *CameraAlarm GetAction.vi*

Gets the alarm actions (Not valid if Alarm Type is Batch Alarm).

10483003;a1

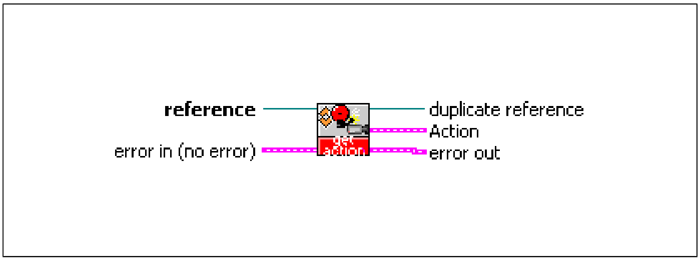


Figure 6.75 Connector Pane

Figure 6.76 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	
	Action	
	TF	**DisableShutter** Shutter will be disabled while alarm is active
	TF	**FreezeImage** Image will be frozen while an alarm is activated. Not implemented!
	TF	**StoreImage** Will store image when alarm is active
	TF	**TrigMarkedImage** Puts camera trigger information into the images when alarm becomes active.
		MarkImage
	FTP/Mail	
	TF	**MailImage** Alarm image is mailed to the email address specified in "CameraAlarm SetConfiguration"
	TF	**MailMovie** Alarm movie is mailed to the email address specified in "CameraAlarm SetConfiguration"
	TF	**MailResult** The alarm result is mailed to the email address specified in "CameraAlarm SetConfiguration"
	TF	**FTPImage** Image is transferred to the ftp-server specified in "CameraAlarm SetConfiguration"

	TF	**TPMovie** Movie is transferred to the ftp-server specified in "CameraAlarm SetConfiguration"
	DigitalOutput	Settings for digital output
	TF	**Digital Output** Activates the digital output channel when an alarm is activated
	I32	**Channel(1)** The digital output channel
	I32	**Duration(ms)** The pulse length in milliseconds for the digital output. 0=no pulse, constant high level during alarm.

6.39 CameraAlarm SetAction.vi

Sets the Normal Alarm actions (You can't set Alarm actions if the Alarm Type is Batch Alarm).

10483103;a1

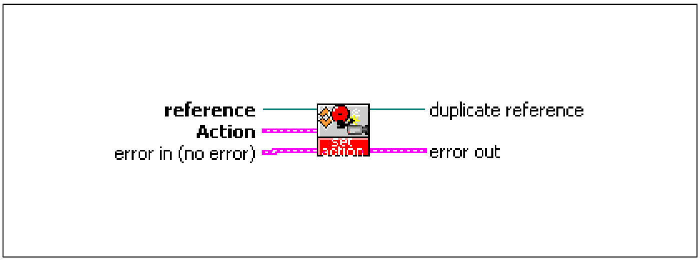


Figure 6.77 Connector Pane

Figure 6.78 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	Action	
		DisableShutter Shutter will be disabled while alarm is active
		FreezeImage Image will be frozen while an alarm is activated. Not implemented!
		StoreImage Will store image when alarm is active
		MarkImage
	FTPMail	
		MailImage Alarm image is mailed to the email address specified in "CameraAlarm SetConfiguration"
		MailMovie Alarm movie is mailed to the email address specified in "CameraAlarm SetConfiguration"
		MailResult The alarm result is mailed to the email address specified in "CameraAlarm SetConfiguration"
		FTPImage Image is transferred to the ftp-server specified in "CameraAlarm SetConfiguration"
		FTPMovie Movie is transferred to the ftp-server specified in "CameraAlarm SetConfiguration"
	DigitalOutput	Settings for digital output

	TF	**Digital Output** Activates the digital output channel when an alarm is activated
	I32	**Channel(1)** The digital output channel
	I32	**Duration(ms)** The pulse length in milliseconds for the digital output. 0=no pulse, constant high level during alarm.
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	

6.40 CameraAlarm GetConfiguration.vi

Basic settings for sending alarms with mail or FTP-transfer

Not supported on A20 or A40 cameras

10778803;a1

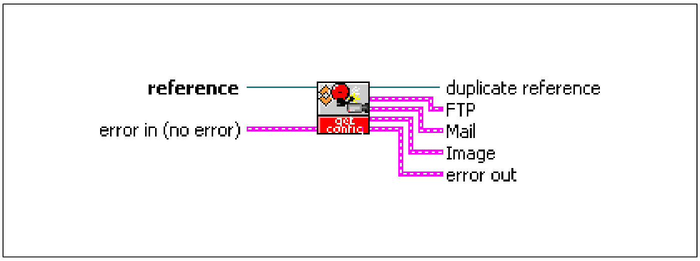


Figure 6.79 Connector Pane

Figure 6.80 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes	
	reference	
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	
	FTP	
		User User is the name you want to use to log on to the FTP server.
		Password Password is the password that authenticates the user-name.
		Host IP FTP host ip address nnn.nnn.nnn.nnn e.g 192.168.0.2
		Active specifies whether the data connection is active or passive. The default is FALSE, which specifies a passive connection.
	Mail	
		MailServerIPAddress nnn.nnn.nnn.nnn where nnn.nnn.nnn.nnn is the mail server ip number
		ReceiverEmailAddress ReceiverName@domain.com
		CameraHostName Mail Client smtp Helo string with the host domain name for the camera. Default is FlirCam
		CameraEmailAddress Reply address provided by the camera (default Alarm@FlirCam). Information about the specific alarm is present in the body of the e-mail

	Image	
		Image Format
		HideGraphics

6.41 CameraAlarm SetConfiguration.vi

Basic settings for sending alarms with mail or FTP-transfer

Not supported on A20 or A40 cameras

10778903;a1

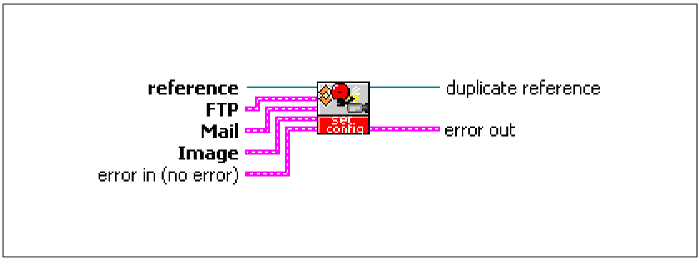


Figure 6.81 Connector Pane

Figure 6.82 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes	
	reference	
	FTP	
		User User is the name you want to use to log on to the FTP server.
		Password Password is the password that authenticates the user-name.
		Host IP FTP host ip address nnn.nnn.nnn.nnn e.g 192.168.0.2
		Active specifies whether the data connection is active or passive. The default is FALSE, which specifies a passive connection.
	Mail	
		MailServerIPAddress nnn.nnn.nnn.nnn where nnn.nnn.nnn.nnn is the mail server ip number
		ReceiverEmailAddress ReceiverName@domain.com
		CameraHostName Mail Client smtp Helo string with the host domain name for the camera. Default is FlirCam
		CameraEmailAddress Reply address provided by the camera (default Alarm@FlirCam). Information about the specific alarm is present in the body of the e-mail
	Image	
		Image Format

		HideGraphics
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	

6

6.42 CameraAlarm GetCondition.vi

Gets the alarm conditions.

10483203;a1

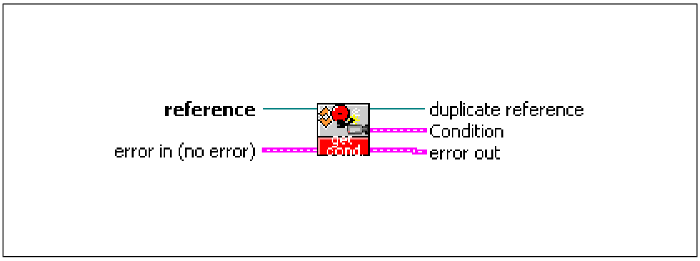


Figure 6.83 Connector Pane

Figure 6.84 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.		
	reference		
	error out error out is a cluster that describes the error status after this VI executes.		
	duplicate reference		
	Condition		
		Source	
			Signal Signal source for the alarm- Analog In n: Analog input channel n- Digital In n: Digital input channel n- Digital Bi-Dir In n: Digital bi-directional input channel n- Spot n, Box n, Line n,Circle n, Diff n. The ID-number "n" must be specified in the n-control.- Ref: Reference temperature- Int. Temp. Sensor: Internal Temperature Sensor
			n "n" is used if Output Source is Spot, Area, Circle or Line
		Threshold The alarm level. Unit depends on source. Not used for digital input.	
		ComparisonType Input signal should be: - Greater(>) than Threshold level, if analog source is used Or High level, if digital source.- Less(<) than Threshold level, if analog source is used Or Low level, if digital source.	
		Hysteresis Hysteresis level. Unit depends on source. Not used for digital input	
		MinDuration Minimum duration while alarm condition is fulfilled.	

6.43 CameraAlarm SetCondition.vi

Sets alarm conditions.

10483303;a1

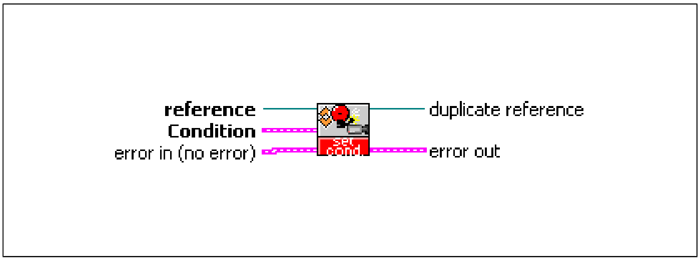


Figure 6.85 Connector Pane

Figure 6.86 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.		
	reference		
	Condition		
		Source	
			Signal Signal source for the alarm- Analog In n: Analog input channel n- Digital In n: Digital input channel n- Digital Bi-Dir In n: Digital bi-directional input channel n- Spot n, Box n, Line n,Circle n, Diff n. The ID-number "n" must be specified in the n-control.- Ref: Reference temperature- Int. Temp. Sensor: Internal Temperature Sensor
		I32	**n** "n" is used if Output Source is Spot, Area, Circle or Line
	DBL	**Threshold** The alarm level. Unit depends on source. Not used for digital input.	
		ComparisonType Input signal should be: - Greater(>) than Threshold level, if analog source is used Or High level, if digital source.- Less(<) than Threshold leel, if analog source is used Or Low level, if digital source.	
	DBL	**Hysteresis** Hysteresis level. Unit depends on source. Not used for digital input	
	DBL	**MinDuration** Minimum duration while alarm condition is fulfilled.	
	error out error out is a cluster that describes the error status after this VI executes.		
	duplicate reference		

6.44 CameraAlarm SetEnable.vi

Enables referenced alarm(if Alarm Type is normal) or all batch alarms(if Alarm Type is Batch Alarm). When an alarm is enabled it...

If Alarm Type is "Normal Alarm": ...begins to search after a valid alarm condition if Batch Alarm is Disabled or Batch Alarm conditions are fulfilled(=true).

If Alarm Type is "Batch Alarm": ...begins to search after a valid alarm (pre-)condition.

10483403;a1

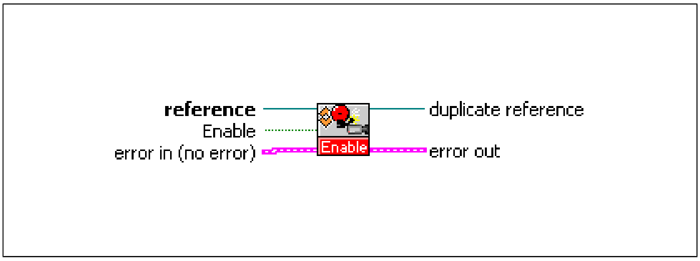


Figure 6.87 Connector Pane

Figure 6.88 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
	Enable
	error out error out is a cluster that describes the error status after this VI executes.
	duplicate reference

6.45 CameraAlarm GetStatus.vi

Displays the alarm status.

10483503;a1

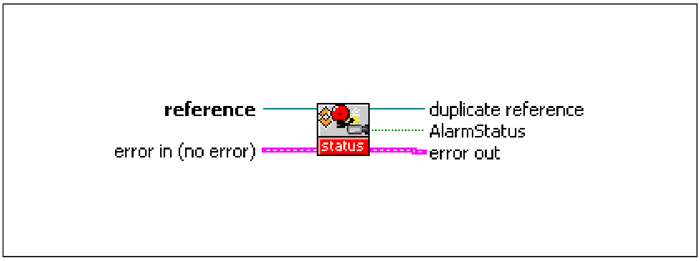


Figure 6.89 Connector Pane

Figure 6.90 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
	error out error out is a cluster that describes the error status after this VI executes.
	duplicate reference
	AlarmStatus

6.46 CameraMeasFunc Create.vi

Creates and initializes a new **CameraMeasFunc** object.

This object should be destroyed using a **CameraMeasFunc** destructor VI.

Used for controlling the measurement functions in following camera type(s):

- ThermaCAM S40/S45/S60/S65
- ThermoVision A20/A40
- ThermaCAM SC640/SC660
- FLIR A320/A300/A310

10483603;a1

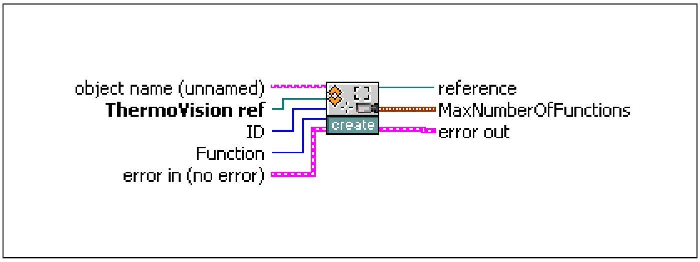


Figure 6.91 Connector Pane

Figure 6.92 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	ThermoVision ref	
	Function The measurement function (spot, box, circle, line, differance, isotherm)	
I32	**ID** Identity number of the measurement function. The ID number starts from one (1) and ends on a camera type specific maximum number(could be zero, if the camera doesn't support the selected function).	
abc	**object name (unnamed)**	
	error out error out is a cluster that describes the error status after this VI executes.	
	reference	
	MaxNumberOfFunctions Maximum allowed measurement functions(0=not allowed)	
	I32	**MaxSpots**
	I32	**MaxBoxes**
	I32	**MaxCircles**
	I32	**MaxLines**

	I32	**MaxDiffs**
	I32	**MaxIsotherms**

6.47 CameraMeasFunc Destroy.vi

Destroys a CameraMeasFunc object created with a CameraMeasFunc constructor VI.

10483703;a1

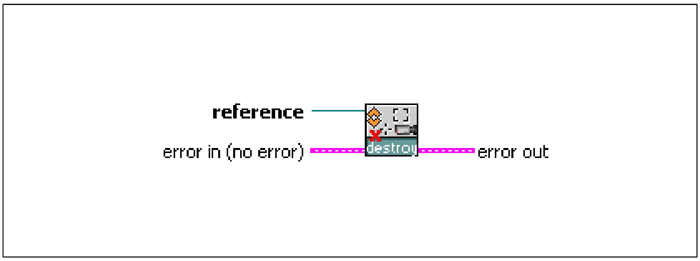


Figure 6.93 Connector Pane

Figure 6.94 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
	error out error out is a cluster that describes the error status after this VI executes.

6.48 *CameraMeasFunc Difference.vi*

Settings for the difference measurement function.

Difference = Input 1 - Input 2

10483803;a1

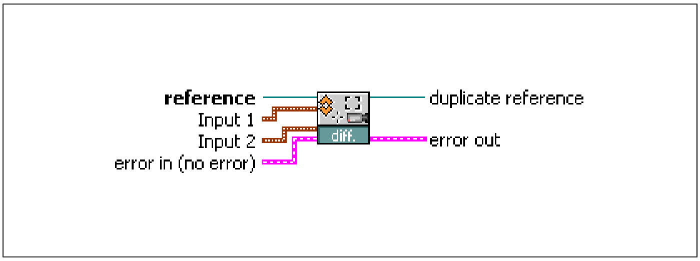


Figure 6.95 Connector Pane

6

Figure 6.96 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	Input 1 - Source: Input source(Spot n, Box n, Circle n, Line n, Diff n, Isotherm n, RefTemp. The ID-number "n" must be specified in the n-control)- Type: Measurement value type(Value, Max, Max Position, Min, Min Position, Average, Standard Deviation, Median).	
		Source Input source(Spot n, Box n, Circle n, Line n, Diff n, Isotherm n, RefTemp. The ID-number "n" must be specified in the n-control)
	I32	**n** The ID number
		Type Type: Measurement value type(Value, Max, Max Position, Min, Min Position, Average, Standard Deviation, Median).
	Input 2 - Source: Input source(Spot n, Box n, Circle n, Line n, Diff n, Isotherm n, RefTemp. The ID-number "n" must be specified in the n-control)- Type: Measurement value type(Value, Max, Max Position, Min, Min Position, Average, Standard Deviation, Median).	
		Source Input source(Spot n, Box n, Circle n, Line n, Diff n, Isotherm n, RefTemp. The ID-number "n" must be specified in the n-control)
	I32	**n** The ID number
		Type Type: Measurement value type(Value, Max, Max Position, Min, Min Position, Average, Standard Deviation, Median).
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	

6.49 CameraMeasFunc Enable.vi

Enables or disables the measurement function.

10483903;a1

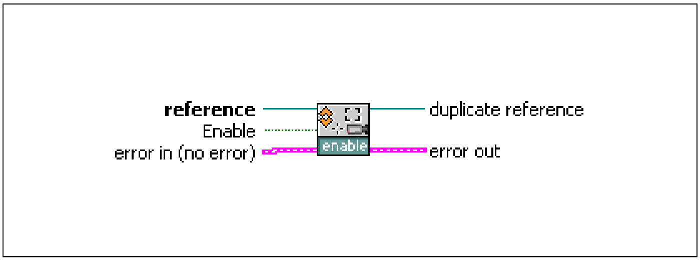


Figure 6.97 Connector Pane

Figure 6.98 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
	Enable Turns On(True) or Off(False) measuerment function.
	error out error out is a cluster that describes the error status after this VI executes.
	duplicate reference

6.50 *CameraMeasFunc Isotherm.vi*

Settings for the isotherm function.

10484003;a1

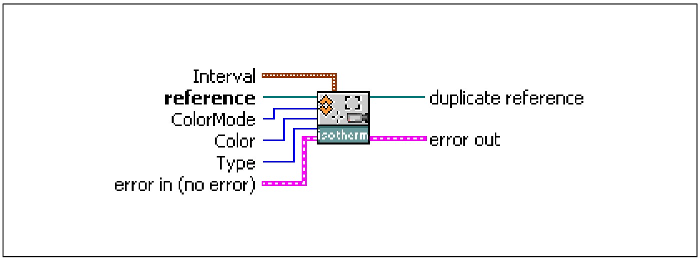


Figure 6.99 Connector Pane

Figure 6.100 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	ColorMode Defines how the isotherm should be represented in the image. Solid, Transparent or Unchanged	
	Color The color to be used in the isotherm(Palette, Palette1, Palette2, Red, Green, Blue, Yellow, Cyan, Magenta, Gray)	
	Type Above, Below or in Interval	
	Interval Temperature range in Kelvin	
	DBL	**High** High temperature value
	DBL	**Low** Low temperature value
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	

6

Publ. No. T559015 Rev. a590 – ENGLISH (EN) – February 29, 2012

6.51 CameraMeasFunc Measurement.vi

Returns the measurement value.

10484103;a1

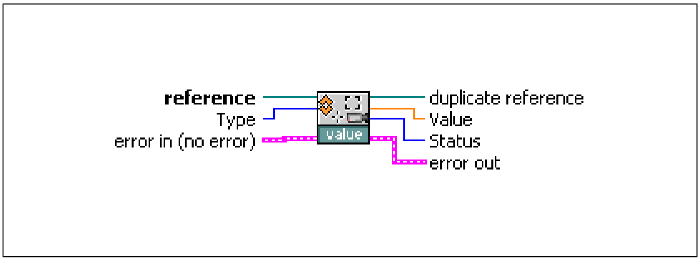


Figure 6.101 Connector Pane

Figure 6.102 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
	Type Measurement value type (Value, Max, Max Position, Min, Min Position, Average, Standard Deviation, Median).Spot, Difference: Only Value is validBox, Circle and Line: All values type except Value are valid.
	error out error out is a cluster that describes the error status after this VI executes.
	duplicate reference
DBL	**Value** The measured value
	Status Status of value(Valid, Out of Calibration Range, Undefined, Outside Image, Greater than, Less than

6.52 *CameraMeasFunc ObjectParameters.vi*

Used for setting object parameters in selected camera measuerement function.

10484203;a1

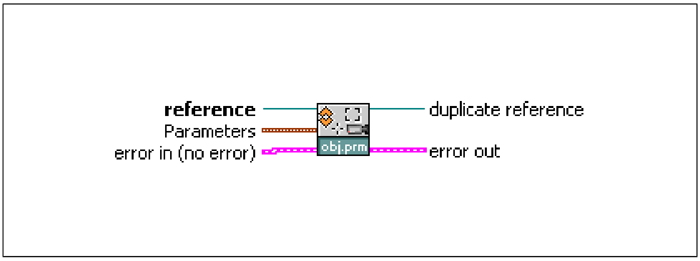


Figure 6.103 Connector Pane

Figure 6.104 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	Parameters	
	DBL	**AmbTemp** Ambient temperature in Kelvin
	DBL	**ObjectDist** Object distance in meters
	DBL	**RelEmissivity** Relative emissivity (0.0-1.0)
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	

6.53 CameraMeasFunc Position.vi

Selects a position for the measurement function in the IR-image.

10484303;a1

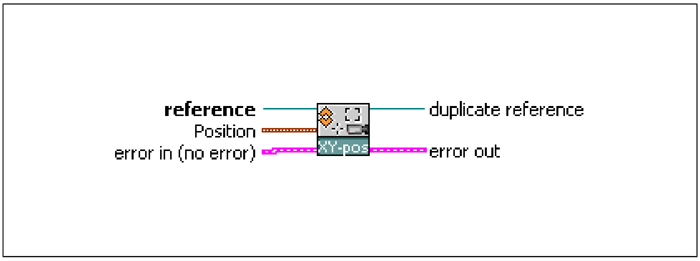


Figure 6.105 Connector Pane

Figure 6.106 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	Position The coordinates for selected measurement function	
	I32	**X1** Left position
	I32	**Y1** Top position
	I32	**X2** Right position(not used if spot function)
	I32	**Y2** Bottom position(not used if spot function)
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	

6.54 CameraPorts Create.vi

Creates and initializes a new CameraPorts object.

This object should be destroyed using a **CameraPorts destructor** VI.

Used for controlling the I/O-ports features in following camera type(s):

- ThermoVision A20/A40
- FLIR A320/A300/A310
- FLIR A325/A320G/A315/SC305/SC325
- FLIR A615/SC645/SC655

10484403;a1

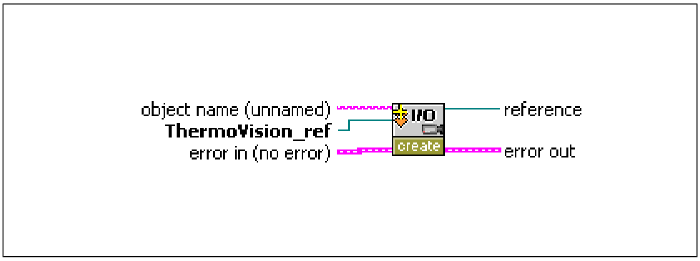


Figure 6.107 Connector Pane

Figure 6.108 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	ThermoVision_ref
	object name (unnamed)
	error out error out is a cluster that describes the error status after this VI executes.
	reference

6.55 CameraPorts Destroy.vi

Destroys a CameraPorts object created with a CameraPorts constructor VI.

10484503;a1

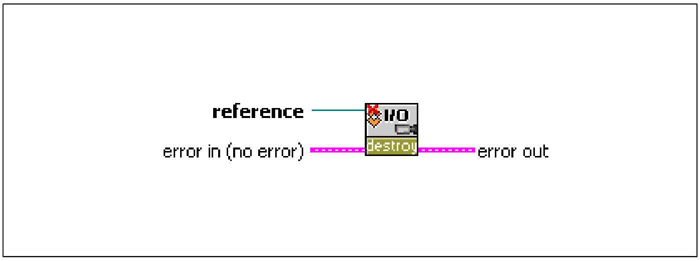


Figure 6.109 Connector Pane

Figure 6.110 Controls and Indicators

	**error in (no error)** error in is a cluster that describes the error status before this VI executes.
	**reference**
	**error out** error out is a cluster that describes the error status after this VI executes.

6.56 CameraPorts AnalogInputConfig.vi

Configures the analog input channel.

10484603;a1

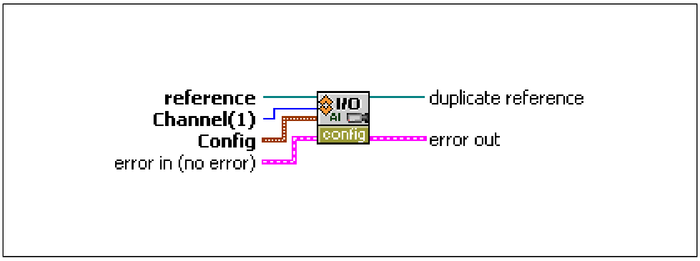


Figure 6.111 Connector Pane

Figure 6.112 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	Config Sets the analog input scale. Used when the analog input channel is read with the AnalogInputReadValue-VI.	
	DBL	**AnalogHigh**
	DBL	**AnalogLow**
I32	**Channel(1)** The analog input channel	
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	

6.57 CameraPorts AnalogInputReadValue.vi

Reads the analog input channel and returns the raw and scaled value.

10484703;a1

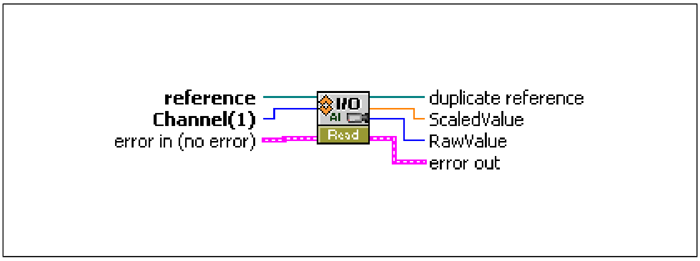


Figure 6.113 Connector Pane

Figure 6.114 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
I32	**Channel(1)** The analog input channel
	error out error out is a cluster that describes the error status after this VI executes.
	duplicate reference
DBL	**ScaledValue** A scaled value(the scale can be configured by using AnalogInputConfig-VI)
I32	**RawValue** A raw value from the AD-converter

6.58 CameraPorts AnalogOutputConfig.vi

Configures the analog output channel.

10484803;a1

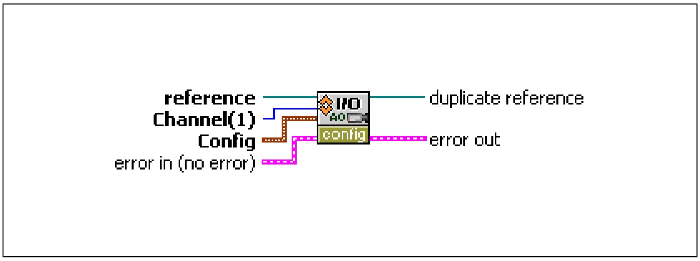


Figure 6.115 Connector Pane

Figure 6.116 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	Config Sets the analog output scale. Used when the analog output value is set with the AnalogOutputReadValue-VI	
	DBL	**AnalogHigh**
	DBL	**AnalogLow**
I32	**Channel(1)** The analog output channel	
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	

6.59 CameraPorts AnalogOutputSignalRoute.vi

Analog outputs can take as a source any measure function value, as well as the value of the internal temperature sensor. It can also relay the analog input value, which in this case will be output as received, without passing by any AD/DA converters.

10484903;a1

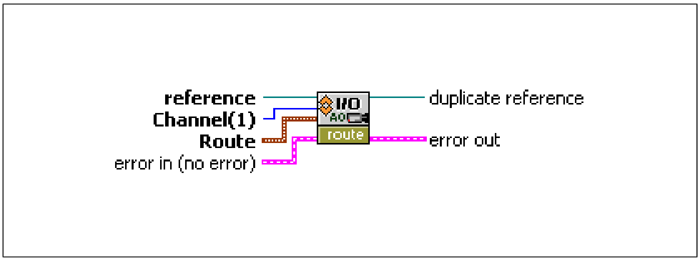


Figure 6.117 Connector Pane

Figure 6.118 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	Route	
		Signal - None: No internal signal is routed - Analog In: The analog input channel is routed to the output channel - Spot n, Box n, Circle n, Line n, Diff n: The measure function value is routed to the output channel. The ID-number "n" must be specified in the n-control. - Int. Temp. Sensor: The internal temperature sensor values is routed to the output channel
		n "n" is used if Output Source is Spot, Area or Line
	Channel(1) The analog output channel	
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	

6.60 CameraPorts AnalogOutputWriteValue.vi

Writes a scaled value to the analog output channel.

10485003;a1

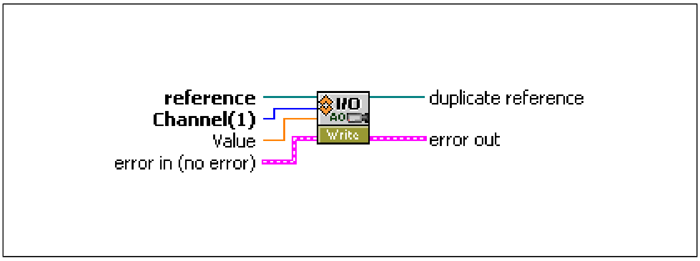


Figure 6.119 Connector Pane

Figure 6.120 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
I32	**Channel(1)** The analog output channel
DBL	**Value** The scaled analog output value(the scale can configured by using the AnalogOutputConfig-VI)
	error out error out is a cluster that describes the error status after this VI executes.
	duplicate reference

6.61 CameraPorts DigitalBiDirConfig.vi

Configures the bi-directional channel direction.

10485103;a1

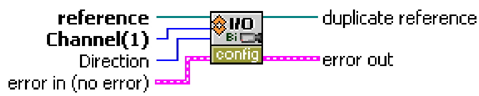


Figure 6.121 Connector Pane

Figure 6.122 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
	Channel(1) The bi-directional channel
	Direction Sets the direction of the channel to input or output
	error out error out is a cluster that describes the error status after this VI executes.
	duplicate reference

6.62 CameraPorts DigitalBiDirReadValue.vi

Reads the bi-directional channel.

10485203;a1

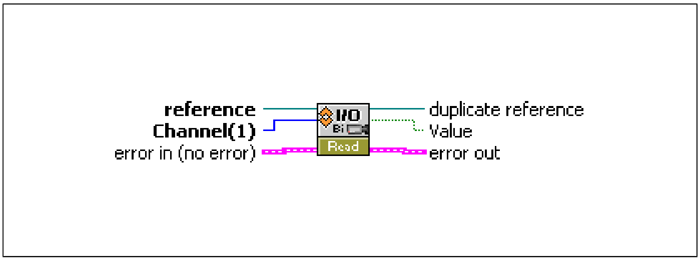


Figure 6.123 Connector Pane

Figure 6.124 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
I32	**Channel(1)** The bi-directional channel
	error out error out is a cluster that describes the error status after this VI executes.
	duplicate reference
TF	**Value** The boolean value from selected channel

6

6.63 *CameraPorts DigitalBiDirSignalRoute.vi*

The bi-directional channel can be used to display some of the internal camera signals. If an internal signal is chosen, the bi-directional channel is set to be output.

10485303;a1

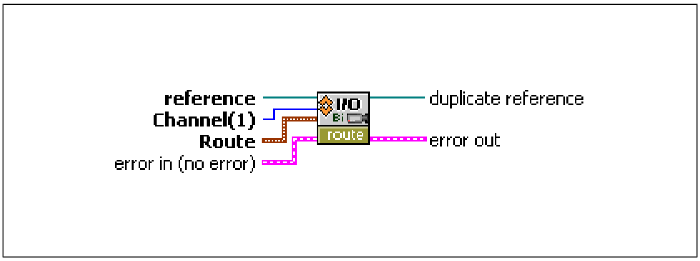


Figure 6.125 Connector Pane

Figure 6.126 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	Route	
		Signal - None: No routing of internal signal to digital output channel - Alarm n: Connects alarm "n" to the bi-directional channel. The Alarm Id-number "n" must be specified in the n-control.
		n
	Channel(1) The bi-directional channel	
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	

6.64 CameraPorts DigitalBiDirWriteValue.vi

Writes to the digital bi-directional channel.

10485403;a1

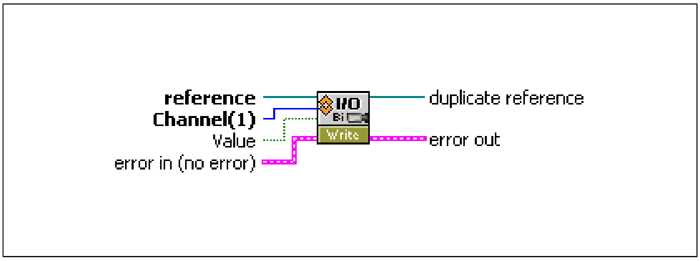


Figure 6.127 Connector Pane

Figure 6.128 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
	Channel(1) The digital bi-directional channel
	Value The value to be written
	error out error out is a cluster that describes the error status after this VI executes.
	duplicate reference

6.65 *CameraPorts DigitalInputReadValue.vi*

Reads the digital input channel.

10485503;a1

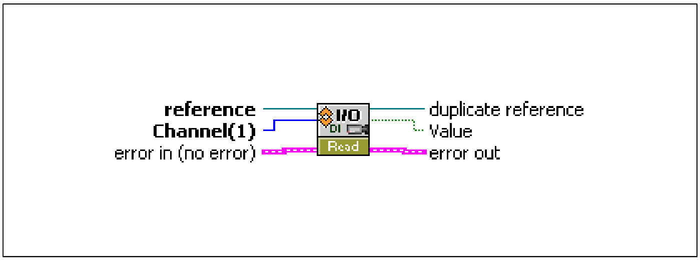


Figure 6.129 Connector Pane

Figure 6.130 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
	Channel(1) The digital input channel
	error out error out is a cluster that describes the error status after this VI executes.
	duplicate reference
	Value The boolean value from selected channel

6.66 CameraPorts DigitalInputSignalRoute.vi

Configures digital input actions

10779003;a1

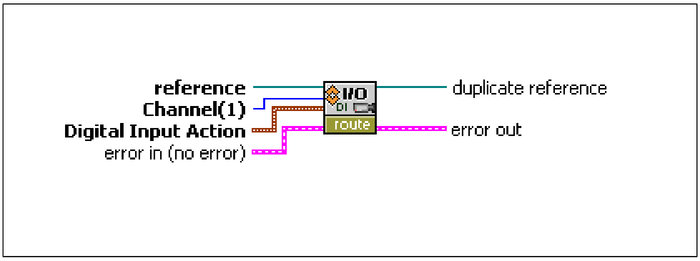


Figure 6.131 Connector Pane

Figure 6.132 Controls and Indicators

	**error in (no error)** error in is a cluster that describes the error status before this VI executes.
	**reference**
	**Digital Input Action**

		Action The following actions and trigger type combinations can be performed with the digital input signal: **0. Off:** No action **1.Marking Images:** Tags streaming image - Mark Image(Rising Flank): On rising flank - Mark Image(Falling Flank): On falling flank - Mark Image(Positive Pulse): During a positive pulse - Mark Image(Negative Pulse): During a negative pulse **2. Start Mark Images:** Start tagging streaming images - Start Mark Image(Rising Flank): On a rising flank - Start Mark Image(Falling Flank): On a falling flank **3. Stop Mark Images:** Stop tagging streaming images - Stop Mark Image(Rising Flank): On a rising flank - Stop Mark Image(Falling Flank): On a falling flank **4. Send Images:** Controls the image stream - Send Images(Positive Pulse): Send images during positive pulse - Send Images(Negative Pulse): Send images during negative pulse **5. Start Send Images:** Start send images - Start Sending Images(Rising Flank): On a rising flank - Start Sending Images(Falling Flank): On a falling flank **6. Stop Send Images:** Stop send images - Stop Sending Images(Rising Flank): On a rising flank - Stop Sending Images(Falling Flank): On a falling flank **7. VSync Input:** Used for syncronizing several cameras, used together with "CameraPorts DigitalOutputRoute.vi" - Syncronize FPA Image(Rising Flank) - Syncronize FPA Image(Falling Flank)
		Trigger Type
I32	**Channel(1)** The digital output channel	
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	

6.67 CameraPorts DigitalOutputSignalRoute.vi

Configures how internal camera functions can be routed to digital output channel.

10485603;a1

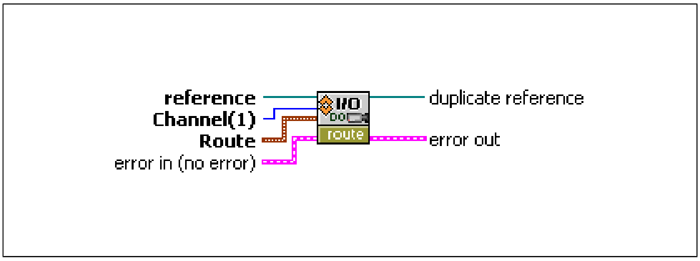


Figure 6.133 Connector Pane

Figure 6.134 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.	
	reference	
	Route	
		Signal - None: No routing of internal signal to digital output channel - Alarm n: Connects alarm "n" to digital output channel - VSync(Rising Flank) : Connects the VSync signal to digital output channel. A rising pulse occurs on digital output when frame start **A20/A40**: Only digital output channel 3 is valid - VSync(Falling Flank) : Connects the VSync signal to digital output channel. A falling pulse occurs on digital output when frame start **A20/A40**: Not used
		n
	Channel(1) The digital output channel	
	error out error out is a cluster that describes the error status after this VI executes.	
	duplicate reference	

6.68 *CameraPorts DigitalOutputWriteValue.vi*

Writes to the digital output channel.

10477503;a1

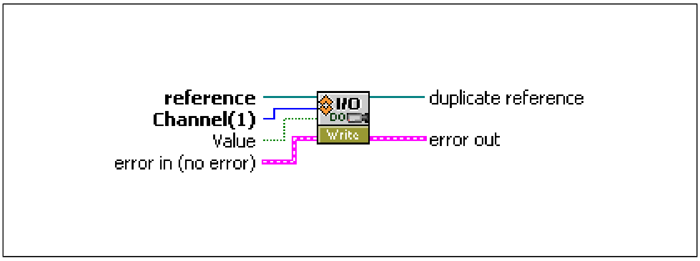


Figure 6.135 Connector Pane

Figure 6.136 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes.
	reference
I32	**Channel(1)** The digital output channel
TF	**Value** The value to be written
	error out error out is a cluster that describes the error status after this VI executes.
	duplicate reference

6.69 ThermoVision RectifyFrameRate.vi

T639446;a1

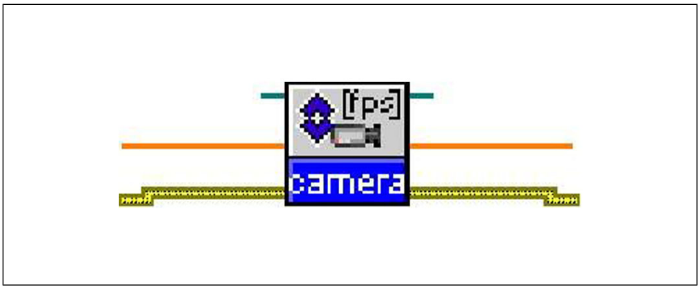


Figure 6.137 Connector Pane

Rounds the input value to nearest significant value. The minimum significant step is 10^3. Solves the discrepancy between reported and actual frame rates found in some discontinued cameras.

The camera can report an available frame rate with some numeric error (for example, 49.999998 instead of 50). However, the actual value must correspond to the reference data. For example, if 50 fps must be used to set needed frame rate, the value 49.999998 will not work.

Figure 6.138 Controls and Indicators

	error in (no error) error in is a cluster that describes the error status before this VI executes. If error in indicates that an error occurred before this VI was called, this VI may choose not to execute its function, but just pass the error through to its error out cluster. If no error has occurred, then this VI executes normally and sets its own error status in error out. Use the error handler VIs to look up the error code and to display the corresponding error message. Using error in and error out clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.	
	TF	**status** status is TRUE if an error occurred before this VI was called, or FALSE if not. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code.
	I32	**code** code is the number identifying an error or warning. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code. Use the error handler VIs to look up the meaning of this code and to display the corresponding error message.

	abc	**source** source is a string that indicates the origin of the error, if any. Usually source is the name of the VI in which the error occurred.
	reference	
DBL	**FrameRate in** Frame rate or image speed(in Hertz)	
	error out error out is a cluster that describes the error status after this VI executes. If an error occurred before this VI was called, error out is the same as error in. Otherwise, error out shows the error, if any, that occurred in this VI. Use the error handler VIs to look up the error code and to display the corresponding error message. Using error in and error out clusters is a convenient way to check errors and to specify execution order by wiring the error output from one subVI to the error input of the next.	
	TF	**status** status is TRUE if an error occurred, or FALSE if not. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code.
	I32	**code** code is the number identifying an error or warning. If status is TRUE, code is a non-zero error code. If status is FALSE, code can be zero or a warning code. Use the error handler VIs to look up the meaning of this code and to display the corresponding error message.
	abc	**source** source is a string that indicates the origin of the error, if any. Usually source is the name of the VI in which the error occurred.
	duplicate reference	
DBL	**FrameRate out** Frame rate or image speed(in Hertz)	

NOTE: This VI is rarely used and thus is absent in palettes. The VI can be found in the library ThermoVision.llb

6

INTENTIONALLY LEFT BLANK

7 Reference section

This section is a complete description of all the possible Methods, Properties, Actions and Events that are part of the ActiveX which ThermoVision™ LabVIEW® Toolkit is based on.

Due to the fact that some functions are not implemented in ThermoVision™ LabVIEW® Toolkit, application programmers and others will need this information during implementation. This is particularly true for the methods **GetCameraProperty** and **SetCameraProperty**.

7.1 About the camera control

7.1.1 Description

Communications between the LabVIEW code and infrared cameras are provided by an ActiveX contained in the file CamCtrl.ocx.

You can use it to switch camera measurement range, control the camera image filtering, view camera status etc. You can also retrieve images and temperature tables.

7

7.1.2 File names

License file: CamCtrl.lic
Binary load file: CamCtrl.ocx
Type library: CamCtrl.tlb

7.1.3 Interfaces

The ActiveX Control Module contains one interface named **FLIR SDK 2.6 Camera Control**.

7.1.4 Camera control properties, methods and events

This control has methods, properties and events by which you can interact with it. The methods and properties allow you to give commands to the control and to retrieve information from it. The events can tell you that something important has happened, for instance that the user has pressed the disconnect button.

The distinction between methods and properties is quite subtle. Properties can be assigned one single value, which methods normally can not. Methods can take parameters, which properties normally can not. The intermediate case, a property with parameters, exists and is handled by this guide as a method because that is the way in which it is regarded by Visual Basic.

This reference guide lists the properties, methods and events in separate sections, in alphabetic order.

7.1.5 Data types

The data types described in this reference manual are those used by the Microsoft® C/C++ compiler. The Visual Basic interpreter uses other names for those data types. See the table below for compatible data types.

7.2 Properties

C/C++	LabVIEW
Float	SGL
Short	I16
BSTR	String
Variant	Variant
Long	I32
Double	DBL

7

7.2.1 Version Property

Returns the camera control version string.

Syntax:

*version = object.***Version**

Item	Description	Value
object	Control object name	For example **CamCtrl**
Version	BSTR	E.g "FLIR Camera Control, ver 2.100"

7.3 Methods

7.3.1 AboutBox Method

You can use the **AboutBox** method to bring up the **About** box of the camera control.

Syntax:

*Object.***AboutBox**

Item	Description	Value
Object	Control object name	For example **CamCtrl**

7.3.2 Connect Method

Connect with the camera using the specified device, communication interface and port. The port value is not used. You should wait for the camera **Connected** event before acquiring any images or setting/getting properties.

Syntax:

Status = *Object.***Connect(** camera, port, device, interface, server **)**

Item	**Description**	**Value**
Object	Control object name	For example **CamCtrl**
Status	Short Return status	0 = OK If nonzero then connection has failed. Use **GetError** method to convert status code to string.
Camera	Short Camera type	**1** ThermaCAM SC2000/SC3000 **2** ThermoVision 160/320 **3** ThermaCAM SC1000 **4** ThermaCAM S40/S45/S60/S65 **5** ThermoVision A20/A40 **6** Cumulus SC4000/SC6000 **7** ThermaCAM SC640/SC660 **8** FLIR A320/A300/A310 **9** FLIR A325 (A320G)/A315/SC305/SC325 **10** Indigo Merlin **11** Indigo Phoenix **12** Indigo Omega **13** ThermoVision 1000 **14** ThermoVision 900 **15** FLIR GF320/GF309 **16** FLIR T series **17** FLIR A615/A645/SC645/SC655 **18** FLIR Exx series **19** FLIR T620/T640
Port	Short. Port to use for communications interface.	0 = Automatic selection of port

7

Item	Description	Value
Device	Short Image source device	**1** FLIR PCCard **2** ITEX IC-DIG 16 **3** FireWire 16-bit images **4** FireWire 8-bit images **5** Ethernet 16-bit images **6** Ethernet 8-bit images **7** IR-FlashLink **8** iPort/GEV **9** USB Video
Interface	Short. Communications interface type.	0. File Only (None) 1. RS232 (Serial) 2. TCP/IP (Socket) 3. AV/C (Firewire) 4.Gigabit (Pleora) 5. AXIS (Axis HTTP 2.x) 6. UVC (USB video class)
Server	BSTR. Camera network address (IP address)	Required if communications interface is AXIS or if image source device is Ethernet (8-bit images). The network address of a camera can be found by using the camera GUI (only for A20/A40). If device is FireWire™ and communication interface is FireWire™, then you may optionally specify the camera serial number in order to connect to a specific camera on the FireWire™ bus.

7.3.3 Disconnect Method

Disconnects the camera and exits the digital transfer mode. You should always disconnect the camera before shutting down your application.

Syntax:

*Status = Object.***Disconnect( )**

Item	Description	Value
Object	Control object name	For example **CamCtrl**
Status	Short.	Return status code.

7.3.4 DoCameraAction Method

Perform a specific camera action as listed below.

Syntax:

*Status = Object.***DoCameraAction(** Action **)**

Item	Description	Value
Object	Control object name	For example **CamCtrl**
Status	Short	Return status code.
Action	Short Action codes Set recording condition properties (ID 27–40) before starting a recording.	0 = Start recording 1 = Stop recording 2 = Enable recording 3 = Disable recording 4 = Pause recording 5 = Resume paused recording 6 = Single snapshot recording 7 = Show recording settings dialog box 8 = Internal image correction (shutter maneuver) 9 = External image correction 10 = Auto adjust 11 = Clear list of recorded files 12 = Auto focus 13 = Show camera information dialog box 14 = Show device status dialog box 15 = Reload calibration from camera 16 = Simulate recording key trig 19 = Save camera settings file (camera type 4 and 5) 20 = Restore settings file (camera types 4 and 5)

7.3.5 EmissCalc Method

Calculates a new emissivity factor for a specified pixel. The image pixel position and a known temperature for this position are supplied as input parameters. If successful, a new calculated emissivity factor (float) is returned. If unsuccessful, an error code is returned (short).

Syntax:

eps = Object.**EmissCalc(** *X, Y, knownTemp* **)**

Item	Description	Value
Object	Control object name	For example **CamCtrl**
eps	VARIANT	Short. Error code Float. New calculated emissivity value
X	Long	Image pixel x coordinate
Y	Long	Image pixel y coordinate
Known temperature	Float	Known pixel temperature (in Kelvin)

7.3.6 GetAbsLUT Method

This method is now obsolete and should not be used.

7.3.7 GetObjLUT Method

Gets a pixel translation table. The table translates absolute image pixels to object pixels. If successful, a memory block will be allocated for the translation table, and it is the responsibility of the caller to de-allocate the memory. The table size depends on the *tableType* parameter.

7

Syntax:

table = *object*.**GetObjLUT(** tableType **)**

Item	Description	Value
Object	Control object name	For example **CamCtrl**
Table	VARIANT	A 1-dimensional array of single precision float translating absolute image pixels to object image pixels.
tableType	Short	1 = Table size 65536 (16-bit absolute image pixels) 2 = Table size 32768 (15-bit absolute image pixels)

7.3.8 GetCameraProperty Method

Gets specific camera property.

Syntax:

Property = *Object*.**GetCameraProperty(** Id **)**

Item	Description	Value
Object	Control object name	For example **CamCtrl**
Property	VARIANT	Camera property value
Id	Short	Property id. See table below.

ID	R/W	Type	Description
0	R/W	Float	Reflected temperature (in Kelvin) Range 0-5000
1	R/W	Float	Atmospheric temperature (in Kelvin) Range 0-5000
2	R/W	Float	Object distance (in meter) Range 0-10000
3	R/W	Float	Object emissivity. Range 0.01–1.00
4	R/W	Float	Relative humidity. Range 0–0.99
5	R/W	Float	Reference temperature (in Kelvin) Range 0-5000
6	R/W	Float	External optics temperature (in Kelvin) Range 0-5000
7	R/W	Float	External optics transmission Range 0.01-1.00
8	R/W	Float	Low Scale Limit (in Kelvin) Range 0-5000
9	R/W	Float	High Scale Limit (in Kelvin) Range 0-5000
10	R/W	Short	Temperature presentation unit 0 = Celsius 1 = Fahrenheit 2 = Kelvin
11	R/W	Short	Distance presentation unit 0 = Meter 1 = Foot
12	R/W	Short	Measurement range array index
13	R/W	BSTR	Focus motor state ("far", "near" or "stop")
14	R/W	Long	Focus absolute position (range depends on camera type)

ID	R/W	Type	Description
15	R/W	BSTR	Cooler state ("on", "off", "standby", "cooling")
16	R	BSTR	Battery status
17	R/W	BSTR	Current camera palette
18	R/W	Short	Deprecated.
19	R	BSTR	Current filter name
20	R/W	Short	Deprecated.
21	R	BSTR	Camera model name
22	R	Short	Current camera type (see Connect Method for values)
23	R	BSTR	Current lens name
24	R	Short	Device type (see Connect Method for values)
25	R	Short	Communications port (see Connect Method for values)
26	R	BSTR	Video mode ("NTSC" or "PAL")
27	R/W	Short	Recording store condition 0 = User 1 = Highest 2 = Every N:th image 3 = At time interval 4 = At external trig
28	R/W	Short	Recording stop condition 0 = User 1 = After time interval 2 = After N images 3 = At external trig
29	R/W	Short	Recording start condition 0 = User 1 = At absolute time 2 = At external trig
30	R	Short	Recording state 0 = Disabled 1 = Waiting for start 2 = Active 3 = Paused

ID	R/W	Type	Description
31	R/W	Short	Source for recording trigs or image marking trigs. 0 = External device or camera 1 = COM port 2 = LPT port (LPT not supported on NT platforms) 3 = None
32	R/W	Short	Recording trig port Range 1-256
33	R/W	Short	Recording file format 0 = Multiple proprietary image files 1 = Sequence file format (*.seq) 2 = Multiple public image files (*.fpf)
34	R/W	Double	Recording start value. Delay in seconds if start condition = 2. Absolute time in days from midnight, 30 December 1899, if start condition = 1.
35	R/W	Double	Recording store value. Image interval if store condition = 2. Time interval in seconds if store condition = 3.
36	R/W	Double	Recording stop value. Time interval in seconds if stop condition = 1. Number of images to record if stop condition = 2. Delay in seconds if stop condition = 3.
37	R/W	BSTR	Recording file base name. Max 10 characters allowed.
38	R/W	Short	Presentation mode 0 = GetImage disabled during active recording 1 = Enabled
39	R/W	BSTR	Recording directory path. E. g. "C:\Images"
40	R	Array (BSTR)	List of recorded files. Use DoCameraAction method to clear this list.

ID	R/W	Type	Description
41	R/W	Short	Noise reduction ThermaCAM™ SC2000: 0-2 = Off 3-5 = Normal >5 = High ThermoVision™ A series: 0–2 = Off >3 = On ThermaCAM™ S series: 0–2 = None 3–5 = Low >5 = High FLIR A3X0, A3X5/SC3X5, A615/SC6X5: Not supported.
42	R/W	Float	Zoom factor Range 1.00-8.00
43	R/W	Double	Frame rate or image speed (in Hertz)
44	R/W	Short	Deprecated.
45	R	BSTR	Calibration title
46	R	Array (BSTR)	List of measurement ranges (in Kelvin). Use this property to determine the number of available measurement ranges. Use the array index to modify the measurement range (property 12)
47	R/W	Short	Automatic shutter. 0=Off 1=On, managed by camera control software 2=On, managed by embedded camera software (SDK default at Connect).
48	R/W	Short	Deprecated.
49	R/W	Short	Scale or overlay graphics visibility (in camera video signal) 0 = Not visible 1 = Visible
50	R/W	Short	Offset Correction Range -500 to +500

ID	R/W	Type	Description
51	R/W	Short	Aperture Range 1-5.
52	R/W	Short	Reserved for internal use
53	R/W	BSTR	IR Source File (absolute path). Only used if image source is file device.
54	R/W	Short	Tooltip visibility 0 = Off 1 = On
55	R/W	Short	Reserved for internal use
56	R/W	Float	Override calculated transmission with estimated value. Range 0-1.
57	R/W	Long	Recording file index
58	R/W	Short	Image rotation 0 = Normal 1 = Horizontal 2 = Vertical 3 = Diagonally
59	R	BSTR	Recording file extension for single file.
60	R	Array (BSTR)	List of available camera palettes
61	R/W	Short	Camera default initialization. Set before connecting. 0 = Disabled 1 = Enabled (default)
62	R/W	Array	Text comments (only for camera type 0, 4, 5, 9-12)
63	R/W	BSTR	Resource path (only for camera type 4, 5, 10, 11, 12, and 13-15)
64	R/W	Any	Resource value (only for camera type 4, 5, 10, 11, 12 and 13-15)
65	R/W	Long	Camera ID (only for camera type 5 and 11). Range 1–999.
66	R	Long	Image line width (in pixels)
67	R	Long	Image height (number of rows or lines)

ID	R/W	Type	Description
68	R	Short	Image pixel size 0 = 16-bit 1 = 8-bit
69	R	Short	Camera status 0 = Connected 1 = Disconnected 2 = Connecting phase 3 = Connection broken 4 = Disconnecting phase
70	W	Short	DirectX viewer 0 = Off 1 = On 2 = Show property page
71	R/W	Short	Image mode 0 = Signal (default) 1 = Temperature 0.1 K 2 = Temperature 0.01 K
72	R/W	Short	Multicast 0 = Use unicast 1 = Use multicast connection (device type 4 and 6) Must be set before connecting.
73	R/W	Short	Deprecated
75	R	Array (double)	List of available frame rates
76	R	Long	Number of images in current image sequence file
77	R/W	Long	Current image number in image sequence file
78	R/W	Short	1 = Wrap to first image in sequence 0 = Do not wrap at end of sequence
79	R/W	Short	0=No downsample 1=Downsample image (only for camera type 5)
80	R	BSTR	Serial number of camera (from image file or camera)
81	R	Long	Deprecated.

7

ID	R/W	Type	Description
82	R/W	Short	0 = Disable correction 1 = Enable emissivity corr. 2 = Enable distance corr.
83	R/W	Short	1 = Override source object parameters 0 = Source decides object parameters
84	R/W	Short	1 = Override source scale parameters 0 = Source decides scaling parameters
85	R/W	Short	0 = Normal mode 1 = Silent mode (no popups)
86	R	Short	1 = Cameras has autofocus 0 = Camera does not have autofocus
87	R/W	Short	1=Subscribe to resource path 0=Unsubscribe to resource path Resource path is set using property 63.
88	R	DATE	File device: Image file time stamp Image acquisition device: Time stamp of last image acquisition
89	R	Short	File device: Millisecond part of image file time stamp Image acquisition device: Millisecond part of last image acquisition
90	R	Long	File device: Image file trig count Image acquisition device: Current trig count of image source
91	R/W	Short	Camera file format (supported for camera type 5, 11 and 12) 0 = JPEG and non-compressed IR pixel data (56 kB) 1 = JPEG and PNG-compressed IR pixel data (19 kB) 2 = JPEG without IR data (16 kB) 3 = Only IR and pixel data not compressed (40 kB) 4 = Only IR and pixel data PNG-compressed (4 kB)
92	R/W	Short	File naming used when snapshot recordings are performed in the camera. 0 = Use normal naming (base name + index) 1 = Use current date and time in file name

ID	R/W	Type	Description
93	R/W	Long	Image request timeout (in milliseconds)
94	R/W	Long	Timeout (in milliseconds) for property 64

7.3.9 GetError Method

Converts an status code or error code to a formatted error string.

Syntax:

*errorString = object.***GetError(** errCode)

Item	Description	Value
Object	Control object name	For example **CamCtrl**
errorString	BSTR	Error string depending on errCode.
errCode	Short	Return code from other methods.

7

Status or error code	Explanation
0	OK. No error.
-1	Unspecified error
1	Device not present
2	Device busy
3	Device driver missing
4	Device driver must be updated
5	Failed to load device firmware
6	Failed to configure device. Resource conflict
8	Service Control Manager error
9	Failed to establish a camera control connection
10	Control connection closed
11	Cannot allocate image buffer
12	Invalid image
13	Device hardware error
14	Timeout waiting for image

Status or error code	Explanation
15	Camera configuration error
16	Image acquisition aborted
17	Image source is not initiated
21	Failed to initiate device
22	Failed to open or access file
24	File media full
25	File format not recognized
26	Known temperature is too close to reflected temperature
27	Known and shown temperature are on different sides of reflected temperature
28	Requested function not supported
29	Invalid parameter

7.3.10 GetImage Method

Gets a row-oriented image from the camera. If successful, a memory block will be allocated for the image pixels, and it is the responsibility of the caller to deallocate the memory. The image size depends on selected camera type.

Syntax:

Image = *Object*.**GetImage(** imageType **)**

Item	Description	Value
Object	Control object name	For example **CamCtrl**
Image	VARIANT	One of the following: ■ A 2-dimensional array with image pixels. Pixel format depends on the imageType parameter. ■ Return error code (short)
imageType	Short	0 = Absolute image pixels (16-bit unsigned integer) 2 = Object signal pixels (single precision float) 3 = Temperature pixels (single precision float) 4 = Relative temperature pixels (8-bit unsigned integer)

7.3.11 GetImages Method

Gets a sequence of images from the camera. An array of image pointers is supplied to the method. It is the responsibility of the caller to allocate the space needed for each image. The image structure is row-oriented. Use the properties for image width (66) and height (67) in order to calculate the space needed for each image. A camera event (**Image captured** event) is sent when the first image has been captured.

Syntax:

Status = *Object.***GetImages(** imageType, imageArray, imageSize **)**

Item	Description	Value
Object	Control object name	For example **CamCtrl**.
Status	VARIANT	Return code.
imageType	Short	0 = Absolute image pixels (16-bit unsigned integer) 2 = Object signal pixels (single precision float) 3 = Temperature pixels (single precision float) 4 = Relative temperature pixels (8-bit unsigned integer)
imageArray	VARIANT (array of 32-bit pointers)	Pointer array of pre-allocated images. Each image consists of at least imageSize bytes.
imageSize	Long	Size in bytes of each image

7.3.12 GetLUT Method

Gets a temperature translation table. The table translates absolute image pixels to temperature. If successful, a memory block will be allocated for the translation table, and it is the responsibility of the caller to deallocate the memory. The table size depends on the *tableType* parameter. The 8-bit LUT depends on the camera temperature scale low and high properties.

Syntax:

Table = *object.***GetLUT(** tableType **)**

Item	Description	Value
Object	Control object name	For example **CamCtrl**

Item	Description	Value
Table	VARIANT	A 1-dimensional array of single precision floats translating absolute image pixels to temperature (in Kelvin).
tableType	Short	0 = Table size 256 (8-bit pixels) 1 = Table size 65536 (16-bit pixels) 2 = Table size 32768 (15-bit pixels)

7.3.13 MLGetImages Method

Get array of images. This method is especially designed for the MATLAB example code.

Syntax:

imageArray = *Object*.**MLGetImages(** *imageType, imageWidth, imageHeight, image* **)**

Item	Description	Value
Object	Control object name	For example **CamCtrl**
imageArray	VARIANT	Array of images with requested pixel type
imageType	Short	0 = Absolute image pixels (16-bit unsigned integer) 2 = Object signal pixels (single precision float) 3 = Temperature pixels (single precision float) 4 = Relative image pixels (8-bit unsigned integer)
imageWidth	Short	Image width in pixels
imageHeight	Short	Image height in pixels
image	Short	Number of images to record

7.3.14 SetCameraProperty Method

Set specific camera property.

SEE ALSO: For details about which properties can be modified, see section 7.3.8 – GetCameraProperty Method on page 112

Syntax:

Status = *Object*.**SetCameraProperty(** Id, Property **)**

Item	Description	Value
Object	Control object name	For example CamCtrl.
Status	Short	Return code
Id	Short	Camera property id **SEE ALSO:** For details about which properties can be modified, see section 7.3.8 – GetCameraProperty Method on page 112
Property	VARIANT	Camera property value

7.3.15 SetImage Method

Sets an image with absolute image pixel data. The image should be row-oriented.

Syntax:

Status = *Object*.**SetImage(** image **)**

Item	Description	Value
Object	Control object name	For example **CamCtrl**
Image	VARIANT	A 2-dimensional array with absolute image pixels in 16-bit unsigned integer format.
Status	Short	Return code

7.3.16 SetEmissMap Method

Set an image emissivity correction map. The correction map should be row-oriented and of the same size as the current image size. Use property 82 to enable or disable the correction function. The emissivity correction is applied when a temperature image is requested and the source image is a 16-bit signal image. The emissivity correction can be used when your source image has objects with highly varying emissivity factors.

Syntax:

Status = *Object*.**SetEmissMap(** Map **)**

Item	Description	Value
Object	Control object name	For example **CamCtrl**
Status	Short	Return code

Item	Description	Value
Map	VARIANT	A 2-dimensional array with emissivity values in float format, one for each image pixel. The individual emissivity values must be in the range 0.01 to 1.00.

7.3.17 SetDistanceMap Method

Set an image distance correction map. The correction map should be row-oriented and of the same size as the current image size. Use property 82 to enable or disable the correction function. The distance correction is applied when a temperature image is requested and the source image is a 16-bit signal image. The distance correction can be used when your source image has objects with varying distance factors. This method will overwrite any existing emissivity correction map.

Syntax:

*Status = Object.***SetDistanceMap(** Map **)**

Item	Description	Value
Object	Control object name	For example CamCtrl.
Status	Short	Return code
Map	VARIANT	A 2-dimensional array with distance values in float format, one for each image pixel. The individual distance values must be in the range 0 to 9999 (meters).

7.3.18 SubmitCamCommand Method

Submits a user command to the camera. The response from the camera is returned in the **CamCmdReply** event. Do not use this method for camera types 4, 5, 10, 11 and 12. Camera types 4, 5, 10, 11 and 12 must use properties 63 and 64 for user commands.

Syntax:

*Status = Object.***SubmitCamCommand(** cmd **)**

Item	Description	Value
object	Control object name	For example **CamCtrl**
status	Short	Return code
cmd	BSTR	Camera command string

7.3.19 ToTemperature Method

Converts a given absolute pixel value to temperature in Kelvin.

Syntax:

*Temperature = Object.***ToTemperature(** abspix, eps **)**

Item	**Description**	**Value**
Object	Control object name	For example **CamCtrl**
Temperature	Float	Temperature in Kelvin
Abspix	Long	Absolute pixel value
Eps	Float	Emissivity factor (0.0-1.0). If 0 then use image emissivity.

7.4 Events

7.4.1 CameraEvent Event

7

The **CameraEvent** event occurs when a camera connection changes state. Events can also be thrown for camera state changes, which affect the image distribution (8 and 9).

Syntax:

Private Sub *object*_**CameraEvent** ([id **As Short**])

Item	**Description**	**Value**
Object	Control object name	For example CamCtrl

Item	Description	Value
Id	Short. Event identifier	2 = Device is connected 3 = Device is disconnected 4 = Device connection broken 5 = Device reconnected from broken connection 6 = Device is in disconnecting phase 7 = Auto adjust event 8 = Start of shutter operation 9 = End of shutter operation 10 = LUT table updated 11 = Recording conditions changed 12 = Image captured 13 = All camera settings re-trieved 14 = Frame rate table available (see property 75) 15 = Frame rate change complet-ed (after setting property 43) 16 = Measurement range table available (see property 46) 17 = Measurement range change completed (after setting property 12) 18 = Image size has changed

7.4.2 CamCmdReply Event

The **CamCmdReply** event occurs when the camera control receives a response from a user command issued from the **SubmitCamCommand** Method.

Syntax:

Private Sub *object*_**CamCmdReply** ([response **As Bstr**])

Item	Description	Value
object	Control object name	For example CamCtrl
response	Command response	Response from call to method SubmitCamCommand

7.4.3 ResourceChanged Event

The **ResourceChanged** event occurs when the camera control receives a resource-changed notification from the camera. An active resource subscription is required in order to receive this event. First setting a resource path using property 63 and then

activating the subscription by setting property 87 will establish an active subscription. This event can only be fired for camera type 5, 11 and 12 using a TCP/IP communications interface.

Syntax:

Private Sub *object*_**ResourceChanged** ([resource **As Bstr**], [value **As Variant**])

Item	**Description**	**Value**
object	Control object name	For example CamCtrl
resource	BSTR	Resource path
value	VARIANT	New resource value

8 Using ThermoVision™ LabVIEW® Toolkit

This section gives the general guidelines you must follow when you use the ThermoVision™ LabVIEW® Toolkit to create a LabVIEW® VI that acquires images from a FLIR Systems IR camera and analyzes the true temperatures in them.

8.1 Guidelines for creating a VI

The following four steps have to be included in your application in order to do temperature measurements:

Step	Action
1	Open connection
2	Configure parameters
3	Image acquisition & temperature conversion
4	Close connection

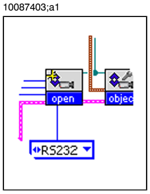
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Begin with calling the ThermoVision Open.vi to create a connection to the IR camera.
10087403;a1


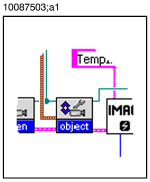
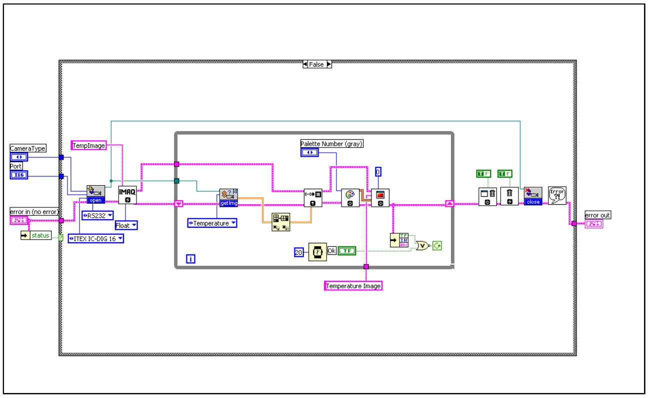
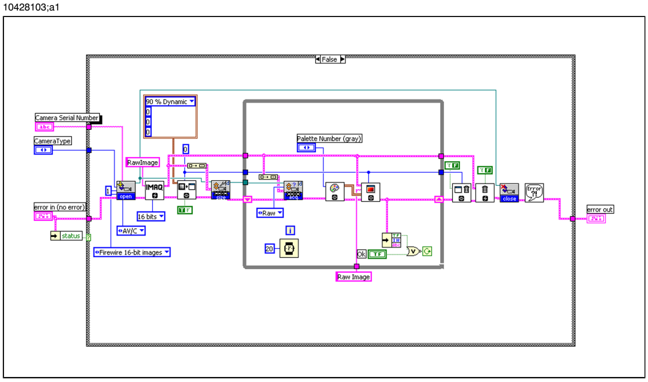
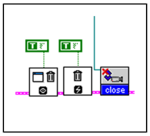
Figure 8.1 ThermoVision Open.vi</td></tr>
<tr><td>2</td><td>Call the configuration VIs to set the different camera parameters (ThermoVision SetObjectParameters.vi, ThermoVision SetDisplayParameters.vi, ThermoVision SetCameraParameters.vi).
10087503;a1


Figure 8.2 ThermoVision SetObjectParameters.vi</td></tr>
</table>

Step	**Action**
3	FireWire/Ethernet: To acquire the images, the ThermoVision GetImage.vi is used. When calling this VI, the type of image is also specified, *i.e.* raw pixel image, absolute pixel image, or temperature pixel image. To convert the two-dimensional image array to IMAQ Vision format, it has to be rotated 90° by using the Transpose 2D-array.vi, after which is relayed to the IMAQ ArrayToImage.vi. 10546203;a1  **Figure 8.3** Image acquisition with FireWire™ or Ethernet
4	FireWire/Ethernet: For faster image acquisitions or burst acquisition from FireWire™ or Ethernet, the ThermoVision Digital GetImages.vi is used. When calling this VI, the type of image is specified, i.e. raw pixel image, absolute pixel image, or temperature pixel image. Before any acquisitions can be made you must pre-allocate image memory with ThermoVision Digital SetImageSize.vi. 10428103;a1  **Figure 8.4** Use SetImageSize and GetImages for higher image acquisition speed, with FireWire™ or Ethernet

Step	Action
5	Before exiting the application, used resources have to be closed in order to prevent memory-resident temporary files, and to make sure certain hardware resources will be available for other applications (*e.g.* the serial port). This is done by calling the IMAQ Close.vi (closing the frame-grabber), IMAQ WindClose.vi (closing the image window), IMAQ Dispose.vi (purging the image memory), and ThermoVision Close.vi (closing the IR camera). 10546303;a1  **Figure 8.5** Closing resources

8.2 The importance of camera calibration

The internal temperature compensation triggers a shutter maneuver whenever a change of the internal temperature is greater than > |0.5 °C| or when more than 15 minutes have elapsed (whichever occurs first). This shutter maneuver is done in order to maintain the measuring accuracy and a correct image. This function can disturb any image acquisition an should be under user control to prevent that the image is NOT obstructed when the important object is within the image.

NOTE: Temperature compensation can NOT be permanently shut off!

Another way to have control over the problem is to shut off the camera's internal temperature compensation and invoke and use the CameraAction VI. The necessity to perform shutter maneuvers is an inherent behavior of a bolometric detector and especially if the user wants to maintain stable video imaging and an high measuring accuracy.

8.3 Configuration parameters

8.3.1 Temperature range and filter strings

Each camera is factory-calibrated to acquire one or more of several different temperature ranges. Each one may or may not allow the use of the camera's spectral filter. The spectral filter blocks out certain infrared wavelengths, altering the upper and lower limits of a factory-calibrated range. Using the filter allows the optimized range to be higher since less infrared radiation reaches the detector.

For example, the limits of a specific factory-calibrated range might be -40-+120 °C, NOF (no filter), but +350-+1500 °C with filter. Each element in the array of strings returned by GetCameraParameters.vi describes one of the range/filter state combinations available on the camera.

The Measurement Range in the Camera Parameters cluster returned by GetCameraParameters indicates the combination currently selected for the camera.

8.3.2 Display parameters

The values in the Display Parameters cluster determine the range of temperatures to acquire and the scale in which to report them.

8.3.2.1 HiScale & LoScale

Has NO effect in the Digital LabVIEW® environment but affects how the video image is displayed. The values must be within the selected temperature range and filter selected.

- *HiScale* sets the maximum temperature displayed in the video image.
- *LoScale* sets the minimum temperature displayed in the video image.

8.3.3 Object parameters

FLIR Systems infrared camera measures and images infrared radiation emitted from an object. The fact that radiation is a function of an object's surface temperature makes it possible for the camera to calculate and display this temperature.

However, the radiation measured by the camera depends not only on the object's temperature but also on its emissivity - its ability to emit radiation. Radiation also originates from the surroundings and is reflected in the object. The radiation coming from the surface of the object and the reflected radiation are also influenced by the absorption of the atmosphere.

To measure temperature accurately, it is therefore necessary for the camera to compensate for the effects of a number of different radiation sources. It does so automatically, based on object parameters that indicate the extent of these other factors.

In the Object Parameters cluster, you supply the values used by the camera to calculate the correct temperatures on the images it captures. Initially, these controls are set to their default values. It is likely that you will need to change all of them.

8.3.3.1 Distance

This is the distance, in meters, between the object and the front lens of the camera. The camera uses this value to correct for the fact that radiation is being absorbed between the object and the camera and the fact that transmittance decreases with distance.

8.3.3.2 *Emissivity*

This is the most important parameter to set correctly. The value indicates the ability of a surface to emit radiation. Valid values are from 0.1 to 1.0. Zero indicates no ability to emit radiation. Normal range is from 0.1 to 0.95.

Here are two methods of establishing the emissivity of the surface you want to image. Both work well if the ambient temperature is not too close to the temperature of the surface. If the surface temperature and ambient temperature are close, any emissivity value will work as well as any other.

NOTE: Make sure the other object parameters are set correctly before you begin either of these procedures.

Setup for the emissivity procedures

Both methods require that you are acquiring live images from the camera, converting them to temperature images, and can place a point object on the image and use the Light Meter (Point) VI to read its temperature results.

8.3.3.2.1 *Using a thermocouple to set emissivity.*

Step	Action
1	Select a reference point on the surface and measure its temperature using a thermocouple.
2	Place a point object on the image at the reference point.
3	Change the emissivity setting until the temperature at the point object agrees with the thermocouple reading. This is the emissivity value of the reference point.

8.3.3.2.2 *Referencing a known emissivity.*

Step	Action
1	Place tape or paint of a known emissivity onto the object.
2	Place a point object on the image at the location of the tape or paint, set the emissivity to the known value, and make a note of the temperature at that point.
3	Move the point object to a reference point adjacent to the tape or paint and adjust the emissivity until the temperature at that point matches the previous reading. This is the emissivity value of the reference point.

8.3.3.3 *Relative humidity*

The camera can also compensate for the fact that transmittance is affected by relative humidity. For short distances and normal humidity, you can leave this setting at its default value of 50 %. However, to increase accuracy, set this value to the actual relative humidity. Valid values are from 0.0 to 1.0.

8.3.3.4 Ambient temperature (Kelvin)

The temperature of the object's immediate environment. The camera uses this setting to compensate for the radiation reflected in the object. The extent of the reflection is determined by the emissivity of the object. For example, if the object's emissivity is 0.75, then 75 % of the object's radiation detected by the camera indicates the temperature of the object, and 25 % indicates the ambient temperature.

If the emissivity is low, the distance very long, and the object's temperature relatively close to the ambient temperature, it is very important to set the ambient temperature correctly, so the camera can accurately compensate for it.

8.3.3.5 Atmospheric temperature (Kelvin)

Temperature of the atmosphere between the camera and the object. The camera uses this setting to compensate for the radiation emitted from the atmosphere between the camera and the object.

9 Redistribution or building a stand-alone applications

9.1 LabVIEW® VI

When you want to create a stand-alone application or when you want to re-distribute the application with ThermoVision™ LabVIEW® Toolkit, make sure you add the following template VI as a dynamic file in Application Builder:

/LabVIEW/user/ThermoVision/ThermoVision.llb/ThermoVision CamCtrl.vit

10428203;a2

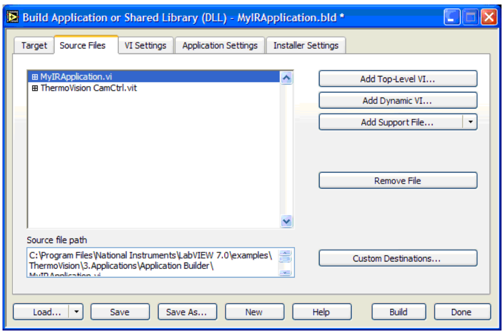


Figure 9.1 Add ThermoVision CamCtrl.vit as Dynamic File in LabVIEW® Application Builder when you want to create an application

9.2 Redistribution

9.2.1 Camera Control Runtime Component

To install the FLIR SDK Camera Control together with your application, please use the supplied installer, which can be found on the CD. The file name of the installer is “ThermoVision™ SDK Runtime.msi”.

9.2.2 Bonjour Core Services 1.0.4

This needs to be installed for FLIR A3XX and A6XX/SC6XX cameras. The file name of the installer is “BonjourSetup.exe”.

9.2.3 FLIR Device Drivers

This needs to be installed for all cameras

9.2.4 Ethernet Bus Drivers

It is recommended that these drivers are installed for FLIR GEV cameras (FLIR A3X5 and A615/SC6X5). The filename of the installer is **FLIR eBus Package.exe**.

INTENTIONALLY LEFT BLANK

10 FireWire™ configuration

10.1 *System parts: ThermaCAM™ S- and ThermoVision™ A-series – FireWire™ interface*

This configuration is used for the following camera models:

- ThermaCAM™ S60
- ThermaCAM™ S65
- ThermaCAM™ S40
- ThermaCAM™ S45
- ThermaCAM™ SC640
- CPA 8200
- CPA 8000
- ThermoVision™ A20 M FireWire™
- ThermoVision™ A40 M FireWire™

10430403;a2

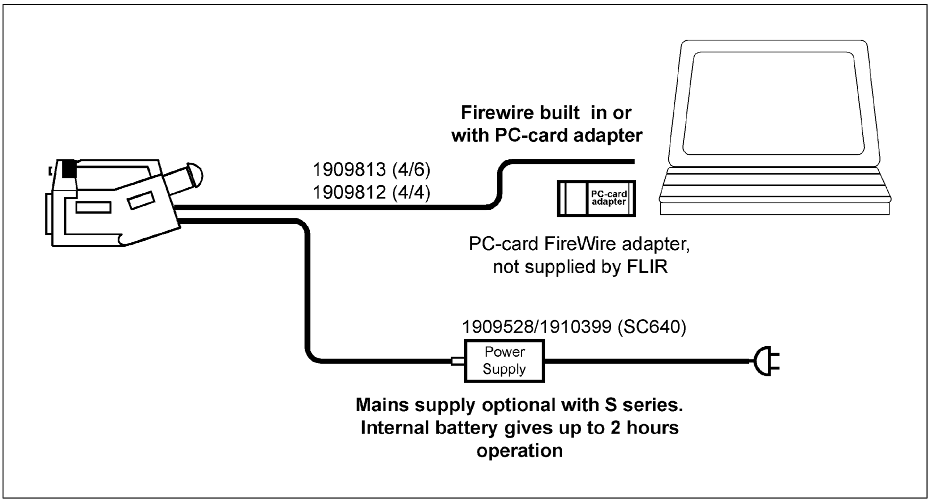


Figure 10.1 ThermaCAM™ S-series – FireWire™ interface & laptop computer

10

10430503;a2

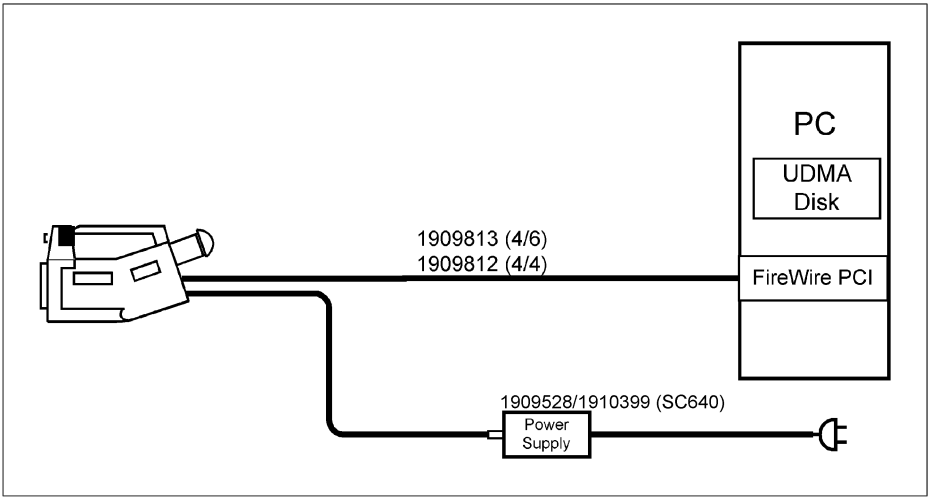


Figure 10.2 ThermaCAM™ S-series – FireWire™ interface & desktop computer

10430603;a1

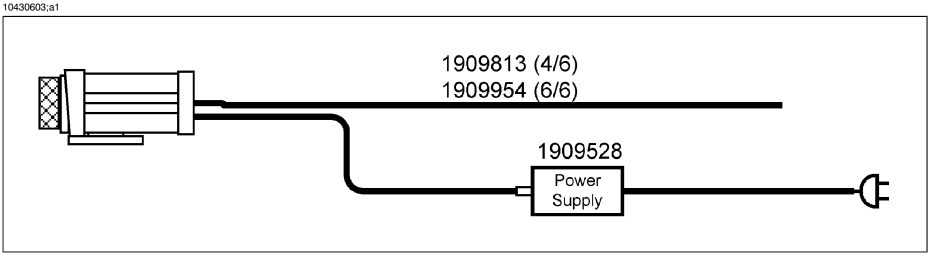


Figure 10.3 ThermoVision™ A-series – FireWire™ interface

Necessary parts:

- A FireWire™ cable, 6 pole/4 pole, max 4.5 m / 14.8 ft.
- A power supply (the camera battery can also be used)
- An optional FireWire™ hub, to be able to add a 6 pole/6 pole extension cable

10.2 Software limitations

The FireWire™ configuration works on Windows® XP,Windows® Vista, and Windows® 7 operating systems.

Full burst rate recording of the cameras is only possible when the target disk is formatted with a NT file system (NTFS) and is fast enough.

10.3 PC recommendations

To get a reasonably high performance you should have a Pentium desktop computer with a clock rate of 1 GHz or more.

If IDE Ultra DMA/100 is supported on your computer, then there is no need for striped SCSI disks. There are such disks that are so fast that you don’t need to use striped volumes. We now recommend 7200 RPM 30 GBUltra DMA/100 disks (or better).

If you have a serial ATA disk (SATA) that is even better.

The FireWire™ adapter in the PC must conform to the 1394a-2000 specifications and must support bus speeds up to 400 Mb/s.

The amount of memory in the PC should be at least twice the Microsoft® recommendation.

Our recommendation is to set the color palette to 256 colors. TrueColor gives a lower frame rate, but better color fidelity.

10.4 Installing the FireWire™ camera driver software

10.4.1 General instructions

Step	Action
1	If needed, insert a FireWire™ Adapter into a free PCI bus slot on your desktop PC. The adapter must conform to the IEEE-1394a-2000 specification and be capable of supporting a serial bus speed of 400 Mb/s. Windows® should automatically detect the new hardware and ask for its drivers, which are supplied by the FireWire™ Adapter manufacturer
2	Disable any sensitive equipment (especially disks), that you may already have connected to your FireWire™ adapter.
3	Switch on the camera and check that the camera has the Digital Video Mode setting DCAM before you plug it into your computer. SC640 does not have this setting.
4	Connect the 1394 cable between the camera to the FireWire™ adapter of the PC when the camera is up and running.

10.4.2 Windows Vista and Windows 7

Step	Action
1	Please log in as Administrator during this installation. Windows® should detect the camera, after a while.

10

Step	Action
2	If Windows® displays a **New Hardware Found** Wizard for the device **FLIR ThemaCAM**, select **Locate and install** and **Don't search online**. Either insert the ThermaCAM™ Researcher CD-ROM or select **Don't have the disk** and **Browse my computer** to let Windows® find an appropriate driver on the CD or in your C:\Program Files\Flir Systems\FLIR Device Drivers directory.
3	Please allow Windows® to continue installing the software despite the complaints about the FLIR drivers not being digitally signed. If Windows® refuses to let you do this, please check that the Windows® Update Driver setting (Right-click on My Computer -> Properties -> Hardware tab) is not set to **Never**.
4	Some cameras will also contain a FLIR 1394 Network Adapter device. You can install it in the same way, if you like. ThermaCAM™ Researcher does not use it.

10.4.3 Windows XP

Step	Action
1	Log in as Administrator. Windows® should detect the camera after a while.
2	If Windows® displays a **New Hardware Found Wizard** for the device FLIR ThemaCAM, either let Windows® search for the best available driver or select **Install from a specific location (Advanced)** to let Windows® find an appropriate driver in your C:\Program Files\Flir Systems\FLIR Device Drivers directory.
3	Allow Windows® to continue installing the software despite the complaints about the FLIR drivers not being digitally signed. If Windows® refuses to let you do this, please check that the Driver Signing setting (Right-click on **My Computer** → **My Computer** → **Properties** → **Hardware** tab) is not set to **Block.**

There are cameras capable of supporting other devices, such as the FLIR 1394 Network Adapter and the FLIR USB Adapter. These devices belong to the ThermaCAM™ Connect 3.0 or ThermaCAM™ QuickReport product, which has drivers for them.

10.5 Troubleshooting the FireWire™ installation

To work properly, the FireWire™ configuration needs:

- Microsoft® Windows® XP, Windows® Vista, or Windows® 7
- Direct X 8.1 (or higher)
- A correct **Type of camera** and **Type of connection** setting in the **Select Camera** dialog.
- A 6 (or 4) to 4 FireWire™ cable for ThermaCAM™ S-series
- A 6 (or 4) to 4 FireWire™ cable for ThermaCAM™ SC640-series
- A 6 (or 4) to 6 FireWire™ cable for ThermoVision™ A-series
- A IEEE-1394a-2000 FireWire™ adapter
- A successful installation of the FireWire™ Adapter driver

- A camera equipped for FireWire™ digital output with its digital video mode set to DCAM
- The Driver Signing setting of the Windows® Device Manager should not block unsigned files
- The TCP/IP protocol Automatic Metric setting should not be set
- A successful installation of the FLIR ThermaCAM™ camera driver for each camera used
- A 1GHz (or faster) PC or laptop equipped with a IEEE-1394a-2000 interface capable of serial bus speed of 400 Mb/s.
- Recent updates from Microsoft® and the computer manufacturer
- With Windows® XP Service Pack 2, S series cameras must have filekit 2.2.5 (or higher) and A series cameras must have filekit 1.2.12 (or higher). The FLIR ThermaCAM™ driver must have version 5.20.2600.923 (or higher).
- Administrator rights (or a change in the Local Security Policy) for the users that plug in/out the camera on Windows® XP
- Proper settings if you have a firewall in your computer.
- Acrobat Reader from http://www.adobe.com

Some laptop computers are not equipped with the correct FireWire™ interface. In such cases, a proper FireWire™ interface can be added using a CardBus interface adapter. A desktop PC needs a free PCI slot in order to install a FireWire™ interface card. The FireWire™ connector of your PC may have 4 or 6 pins.

The IEEE-1394a-2000 adapter must be capable of a serial bus speed of 400 Mb/s in order to achieve full real time recording speed (50/60Hz). Even when this is the case, limitations elsewhere in the computer may not allow full speed.

With some laptop chipsets there is a problem cause by too much latency in the C3 power state transition which cause buffer underruns. This can be cured by a change in the Windows® registry. For more information, see Publ. No. T559004, Installation Hints, on the CD-ROM.

Since FireWire™ is a fairly recent addition to the Windows® world, hardware and software weaknesses still plague the technology. We recommend that you visit the Microsoft® Windows® Update website (windowsupdate.microsoft.com) to refresh your software and Windows® drivers, and that you visit the corresponding site of your computer manufacturer to receive its latest updates.

Do not connect other FireWire™ equipment to your computer when you transfer IR images.

Please disable any sensitive equipment (especially disks) that you may already have connected to your FireWire™ adapter before you plug in the camera.

INTENTIONALLY LEFT BLANK

11 Gigabit Ethernet interface configuration

11.1 System parts: Gigabit Ethernet interface

This configuration is used for the following camera models:

- FLIR A3X5/SC3X5
- FLIR A615
- FLIR SC645
- FLIR SC655

10570003;a3

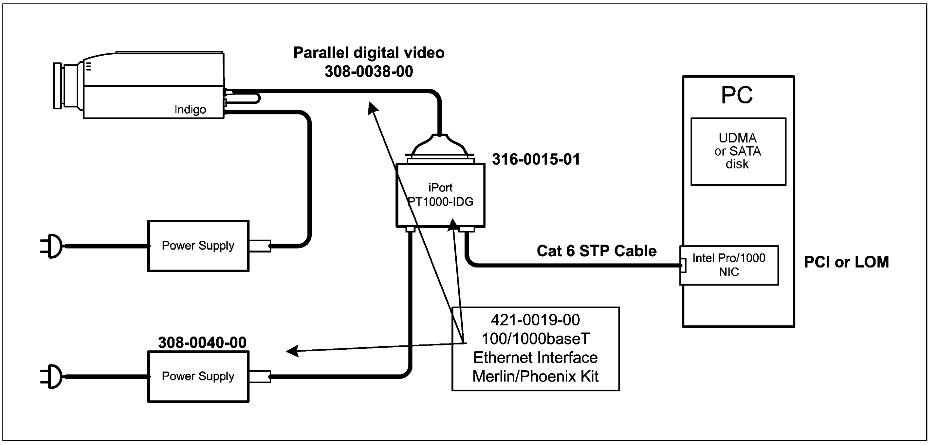


Figure 11.1 Indigo Merlin system parts. LOM = Lan on Motherboard.

11

10570103;a3

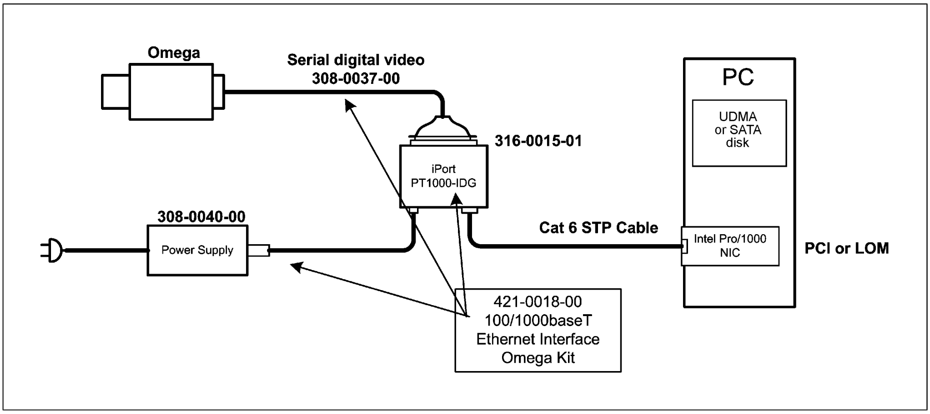
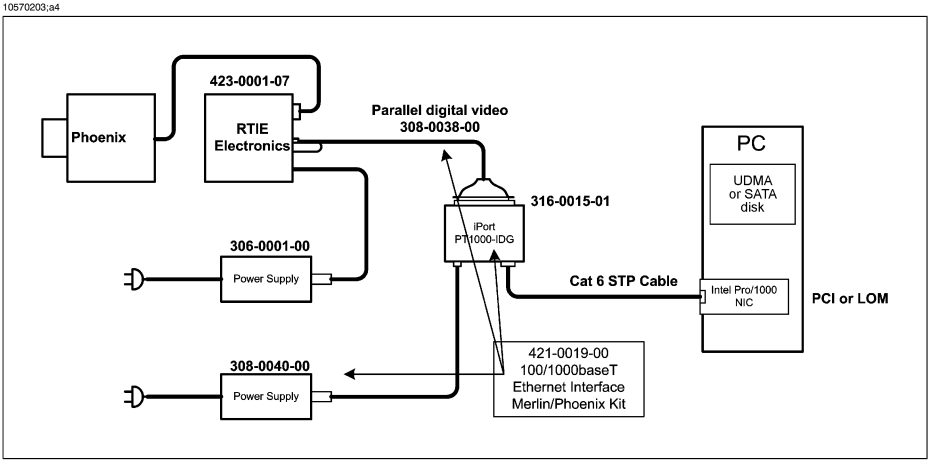


Figure 11.2 Indigo Omega system parts

10570203;a4



11

Figure 11.3 Indigo Phoenix system parts

10772003;a1

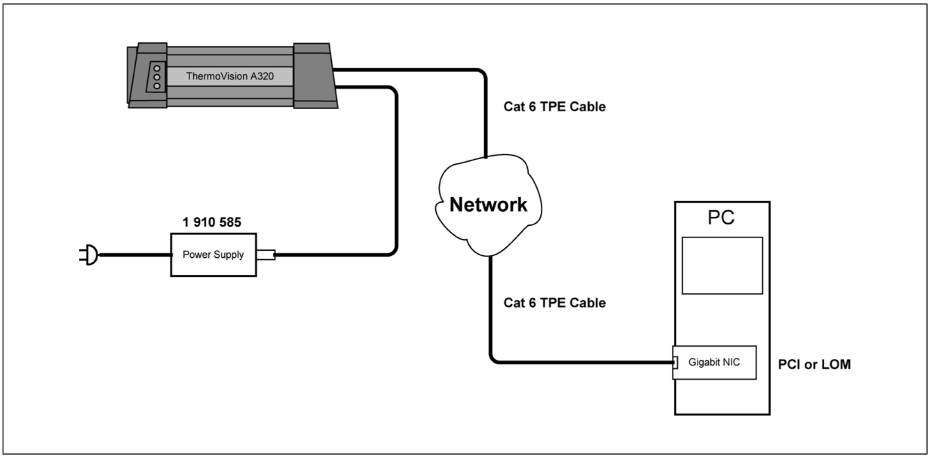


Figure 11.4 FLIR A3X5/SC3X5 and A615/SC6X5 system parts

Necessary parts:

- Intel PRO/1000 compatible Network Interface Card (NIC) in the PC
- A standard CAT6 STP Ethernet cable (up to 100 meters without intervening hardware).
- A desktop computer with a fast UDMA or SATA disk for high speed image storage

11.2 Software limitations

- The Gigabit Ethernet configuration works on Windows® XP, Windows® Vista, and Windows® 7 operating systems (all 32- and 64-bit).
- Full burst rate recording of the cameras is only possible when the target disk is formatted with a NT File System (NTFS).
- Pleora and eBus drivers are only supported on Windows® XP, Windows® Vista, and Windows® 7 operating systems (all 32- and 64-bit).

11

11.3 PC recommendations

Minimum requirements are:

- A Pentium 4 computer with a clock rate of 1.4 GHz or more.
- At least 512 MB RAM.
- Gigabit Ethernet network adapter (either PCI card or LAN on the motherboard).

To achieve burst recording it should have an 7200 rpm Ultra DMA/100 disk formatted with the NT file system (NTFS). Ultra DMA/100 requires Windows® XP, Windows® Vista (32-bit), and Windows® 7 operating systems.

If you have a serial ATA (SATA) disk, that is even better.

11.4 Installing driver software for the Gigabit Ethernet interface

11.4.1 Windows® XP/Vista/7

To take full advantage of the GigaBit interface you need the eBus optimal driver.

To copy the eBus driver suite onto your hard disk, run the installation program present on the CD. The Pleora eBus installation tool is available by running "C:\Program Files\Common Files\Pleora\EbDriverTool.exe".

Note that the eBus optimal driver is only compatible with the Intel PRO/1000 family of network adapters (either a PC network interface card, often referred to as a NIC, or a LAN on the motherboard, often referred to as a LOM). If you have some other kind of network adapter, use the universal driver instead.

The ordinary driver for your network adapter has almost certainly already been installed by Windows. You will have to update the previous installation with the Pleora driver.

You need to log in as administrator (or as a user with administrator rights) to do this.

Please follow these steps to replace your Gigabit network interface driver with the Pleora device driver:

1. First make sure that your computer has been fully updated by **Windows® Update**.
2. Run the Pleora installation tool. The Pleora eBus installation tool is available by running "C:\Program Files\Common Files\Pleora\EbDriverTool.exe".
3. Find your Gigabit Ethernet Adapter and select **Action**.
4. Select the optimal eBus Driver, if you have Intel PRO/1000 adapter, or the universal if you have another adapter. Click **Install**.
5. You may also have to update the new driver.
6. Press Exit and allow the computer to reboot.

You will have to let your computer's Firewall allow your application to access the network to be able to connect to your camera.

If you do not use an Intel PRO/1000 network adapter or do not install the Pleora drivers, the software will still allow you to set an IP number to the interface and try to connect to the camera using the ordinary networking functions of Windows. This connection will not give full speed performance for most cameras. For more information about this, please see section 4.3.

11.5 Troubleshooting the Gigabit Ethernet interface installation

To work properly, the configuration needs:

- Windows® /XP/Vista/7 (32- and 64-bit)
- A functional Gigabit Network Interface Card (NIC). Please note that configurations with more than one Gigabit network interface and optimum driver have not been tested.
- A successful installation of the optimum eBus device driver.
- A Pentium 4 computer with a clock rate of 1.4 GHz (or faster). At least 512 MB RAM.
- Select **Ethernet** as **Type of connection** in the **Select Camera** dialog.
- NetBIOS enabled on the TCP/IP connection to the camera
- Reliable cables and electrical connections. Gigabit Network adapters require shielded CAT6 cables.
- A camera equipped and configured for digital output.
- Proper settings if you have a firewall in your computer.

INTENTIONALLY LEFT BLANK

12 Standard Ethernet interface configuration

12.1 System parts: Standard Ethernet interface configuration

This configuration is used for the following camera models:

- FLIR A3X0

10574003;a1

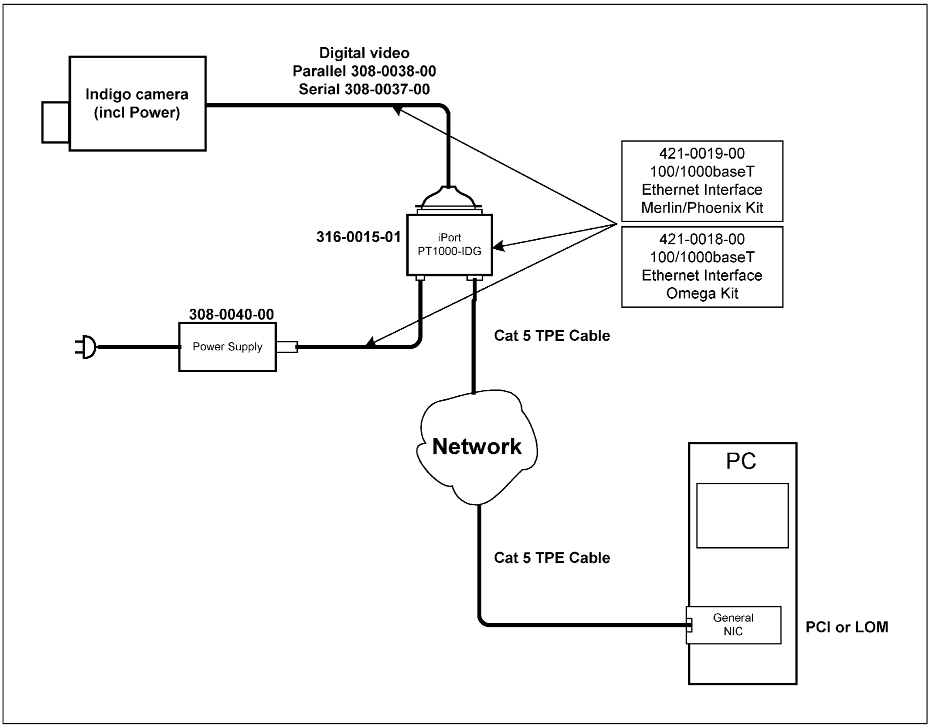


Figure 12.1 Indigo Merlin/Omega/Phoenix system parts. LOM = Lan on Motherboard.

10771903;a1

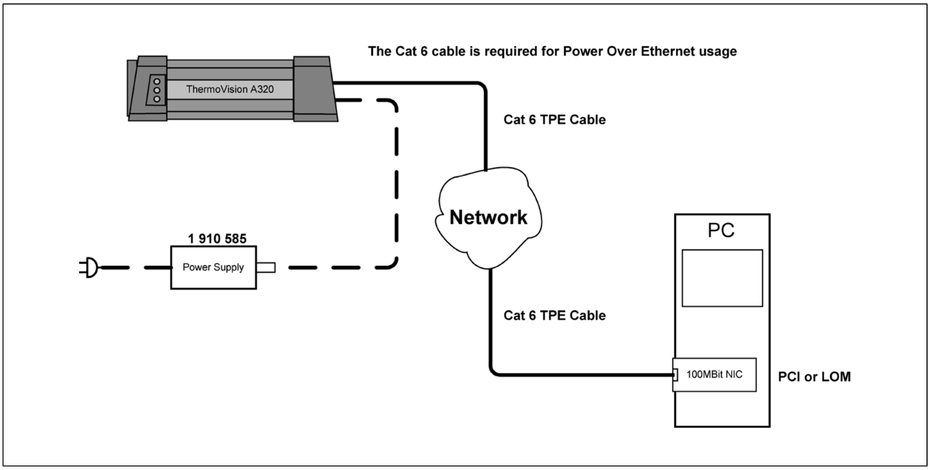


Figure 12.2 FLIR A3X0 system parts. LOM = Lan on Motherboard.

Necessary parts:

- Any network interface supporting at least Fast Ethernet (100 Mbit/s)
- A standard CAT5 Ethernet cable (up to 100 meters without intervening hardware)
- A desktop or laptop computer

12.2 Software limitations

The Ethernet configuration works on Windows® XP (32- and 64-bit), Windows® 7 (32- and 64-bit), and Windows® Vista (32- and 64-bit) operating systems.

Full burst rate recording of the cameras is not possible in this configuration.

12.3 PC recommendations

Minimum requirements are:

- A Pentium 4 computer with a clock rate of 1.4 GHz or more.
- At least 512 MB RAM.
- Any network interface supporting at least Fast Ethernet (100 Mbit/s)

12

12.4 Ethernet bandwidth requirements

It is important to understand that the cameras in this configuration will stream uncompressed digital video data on the network. This will consume a lot of bandwidth and can affect the normal network traffic causing congestions and slow response.

It is strongly recommended that you configure your network so that the digital video packet data does not affect any normal Ethernet traffic. Small local area networks designated for video streaming is recommended. These networks can either be Fast Ethernet or Gigabit Ethernet networks depending on the camera model (see figure below).

Figure 12.3 Network type recommendations

Camera model	Bandwidth requirement	Network recommendation
Merlin (60 Hz)	~100 Mbit/s	Use Gigabit Ethernet
Omega (30 Hz)	~12 Mbit/s	Use Fast Ethernet
Phoenix (60 Hz) Resolution 320 x 256	~100 Mbit/s	Use Gigabit Ethernet
Phoenix (60 Hz) Resolution 640 x 512	~400 Mbit/s	Use Gigabit Ethernet

12.5 Troubleshooting the standard Ethernet interface installation

To work properly, the configuration needs:

- Windows® XP/Vista/7 (32- and 64-bit)
- Any network interface supporting at least Fast Ethernet (100 Mbit/s)
- A Pentium 4 computer with a clock rate of 1.4 GHz or more. At least 512 MB RAM.
- Select **Ethernet** as **Type of connection** in the **Select camera** dialog box
- NetBIOS enabled on the TCP/IP connection to the camera
- Proper settings if you have a firewall in your computer
- Reliable cables and electrical connections
- A camera equipped and configured for digital output
- An IP number assigned to the interface
- An IP number assigned to the camera

INTENTIONALLY LEFT BLANK

13 FLIR Public File image format

13.1 General information

The "xxxx.fpf" files consist of a header followed by a matrix of single precision IEEE floating-point values, each representing one point of the image. A C-style description of the header layout can be found in the header file fpfimg.h, available in the installation directory. The image point values are stored starting from the top left corner, row by row. The FLIR camera control can only save FPF images with floating point temperature values

13.2 Basic data

Char	8 bit	Often represents ASCII characters, may represent an 2's complement 8 bit integer (-128--+127)
Unsigned char	8 bit	8 bit integer number (0--255)
Short	16 bit	16 bit integer (2's complement)
Unsigned short	16 bit	16 bit integer
Long	32 bit	32 bit integer (2's complement)
Unsigned long	32 bit	32 bit integer
Float	32 bit	IEEE floating point number, sign + 23 bit mantissa + 8 bit exponent, Representing numbers in the range +/- 1038
Char[<len>]	Len * 8 bit	ASCII character string, most certainly terminated with the NUL character (= 0)
Int	32 bit	32 bit integer (2's complement)

Multiple byte data types are stored with the least significant byte first.

13.3 The whole header data structure (size 892 bytes)

```
typedef struct
{
   FPF_IMAGE_DATA_T  imgData;
   FPF_CAMDATA_T     camData;
   FPF_OBJECT_PAR_T  objPar;
   FPF_DATETIME_T    datetime;
   FPF_SCALING_T     scaling;
```

13

```
   long  spareLong[32];    /* = 0 */
} FPFHEADER_T;
```

13.4 The image data structure (120 bytes)

```
typedef struct
{
   char fpfID[32];                 /*  "FLIR Public Image Format" */
   unsigned long  version;         /*  = 2 */
   unsigned long  pixelOffset;     /*  Offset to pixel values from
                                       start of fpfID. */
   unsigned short ImageType;       /*  Temperature        = 0,
                                       Diff Temp          = 2,
                                       Object Signal      = 4,
                                       Diff Object Signal = 5, etc */
   unsigned short pixelFormat;     /*  0 = short integer  = 2 bytes
                                       1 = long integer   = 4 bytes
                                       2 = float          = 4 bytes
                                       3 = double         = 8 bytes*/
   unsigned short xSize;
   unsigned short ySize;
   unsigned long  trig_count;      /* external trig counter */
   unsigned long  frame_count;     /* frame number in sequence */
   long  spareLong[16];            /* = 0 */
} FPF_IMAGE_DATA_T;
```

13.5 The camera data structure (360 bytes)

```
/*  String lengths    */
#define FPF_CAMERA_TYPE_LEN   31   /* Camera name string */
#define FPF_CAMERA_PARTN_LEN  31   /* Camera part number string */
#define FPF_CAMERA_SN_LEN     31   /* Scanner serial number string */
#define FPF_LENS_TYPE_LEN     31   /* Lens name string */
#define FPF_LENS_PARTN_LEN    31   /* Lens part number string */
#define FPF_LENS_SN_LEN       31   /* Lens serial number string */
#define FPF_FILTER_TYPE_LEN   31   /* Filter name string */
#define FPF_FILTER_PARTN_LEN  31   /* Filter part number string */
#define FPF_FILTER_SN_LEN     31   /* Filter serial number string */
typedef struct
{
    char camera_name[FPF_CAMERA_TYPE_LEN+1];
    char camera_partn[FPF_CAMERA_PARTN_LEN+1];
    char camera_sn[FPF_CAMERA_SN_LEN+1];

    float camera_range_tmin;
    float camera_range_tmax;

    char lens_name[FPF_LENS_TYPE_LEN+1];
    char lens_partn[FPF_LENS_PARTN_LEN+1];
    char lens_sn[FPF_LENS_SN_LEN+1];
    char filter_name[FPF_FILTER_TYPE_LEN+1];
    char filter_partn[FPF_FILTER_PARTN_LEN+1];
    char filter_sn[FPF_FILTER_SN_LEN+1];
   long  spareLong[16];            /* = 0 */
}FPF_CAMDATA_T;
```

13

13.6 The object parameters data structure (104 bytes)

```
typedef struct
{
   float emissivity;         /* 0 - 1 */
   float objectDistance;     /* Meters */
   float rflTemp;            /* Reflected temperature in Kelvin */
   float atmTemp;            /* Atmospheric temperature in Kelvin */
   float relHum;             /* 0 - 1 */
```

```
   float compuTao;          /* Computed atmospheric transmission */
   float estimTao;          /* Estimated atmospheric transmission */
   float refTemp;           /* Reference temperature in Kelvin */
   float extOptTemp;        /* Kelvin */
   float extOptTrans;       /* 0 - 1 */
   long  spareLong[16];     /* = 0 */
} FPF_OBJECT_PAR_T;
```

13.7 The date and time data structure (92 bytes)

```
typedef struct
{
   int Year;
   int Month;
   int Day;
   int Hour;
   int Minute;
   int Second;
   int MilliSecond;
   long  spareLong[16];            /* = 0 */
} FPF_DATETIME_T;
```

13.8 The scaling data structure (88 bytes)

```
typedef struct
{
   float tMinCam;          /* Camera scale min, in current output */
   float tMaxCam;          /* Camera scale max */
   float tMinCalc;         /* Calculated min (almost true min) */
   float tMaxCalc;         /* Calculated max (almost true max) */
   float tMinScale;        /* Scale min */
   float tMaxScale;        /* Scale max */
   long  spareLong[16];    /* = 0 */
} FPF_SCALING_T;
```

INTENTIONALLY LEFT BLANK

14 About FLIR Systems

FLIR Systems was established in 1978 to pioneer the development of high-performance infrared imaging systems, and is the world leader in the design, manufacture, and marketing of thermal imaging systems for a wide variety of commercial, industrial, and government applications. Today, FLIR Systems embraces five major companies with outstanding achievements in infrared technology since 1958—the Swedish AGEMA Infrared Systems (formerly AGA Infrared Systems), the three United States companies Indigo Systems, FSI, and Inframetrics, and the French company Cedip. In November 2007, Extech Instruments was acquired by FLIR Systems.

T638608;a1

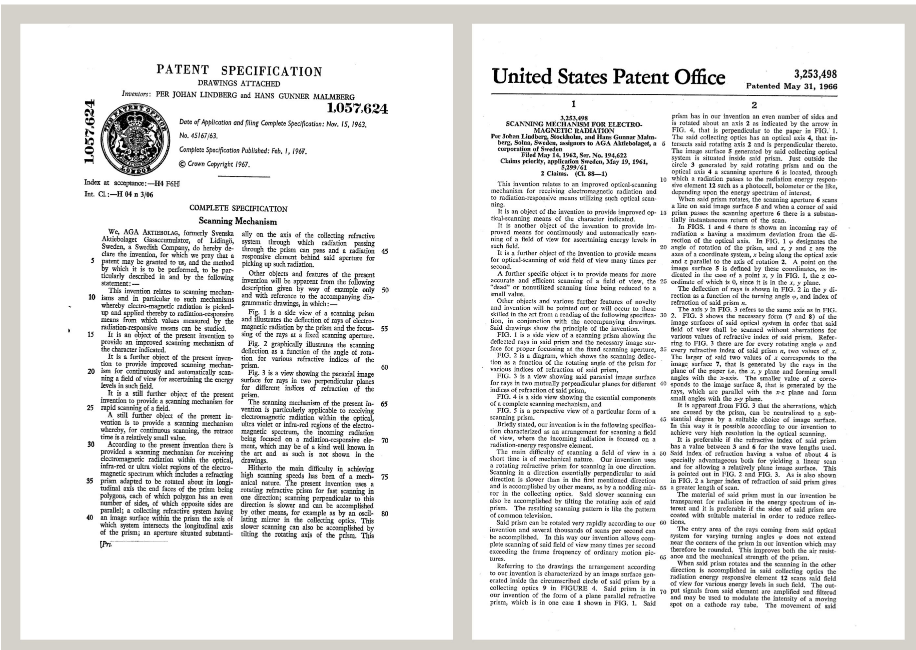


Figure 14.1 Patent documents from the early 1960s

The company has sold more than 200,000 infrared cameras worldwide for applications such as predictive maintenance, R & D, non-destructive testing, process control and automation, and machine vision, among many others.

FLIR Systems has three manufacturing plants in the United States (Portland, OR, Boston, MA, Santa Barbara, CA) and one in Sweden (Stockholm). Since 2007 there is also a manufacturing plant in Tallinn, Estonia. Direct sales offices in Belgium, Brazil,

14

China, France, Germany, Great Britain, Hong Kong, Italy, Japan, Korea, Sweden, and the USA—together with a worldwide network of agents and distributors—support our international customer base.

FLIR Systems is at the forefront of innovation in the infrared camera industry. We anticipate market demand by constantly improving our existing cameras and developing new ones. The company has set milestones in product design and development such as the introduction of the first battery-operated portable camera for industrial inspections, and the first uncooled infrared camera, to mention just two innovations.

10722703;a2

Figure 14.2 LEFT: Thermovision® Model 661 from 1969. The camera weighed approximately 25 kg (55 lb.), the oscilloscope 20 kg (44 lb.), and the tripod 15 kg (33 lb.). The operator also needed a 220 VAC generator set, and a 10 L (2.6 US gallon) jar with liquid nitrogen. To the left of the oscilloscope the Polaroid attachment (6 kg/13 lb.) can be seen. **RIGHT:** FLIR i7 from 2009. Weight: 0.34 kg (0.75 lb.), including the battery.

FLIR Systems manufactures all vital mechanical and electronic components of the camera systems itself. From detector design and manufacturing, to lenses and system electronics, to final testing and calibration, all production steps are carried out and supervised by our own engineers. The in-depth expertise of these infrared specialists ensures the accuracy and reliability of all vital components that are assembled into your infrared camera.

14.1 More than just an infrared camera

At FLIR Systems we recognize that our job is to go beyond just producing the best infrared camera systems. We are committed to enabling all users of our infrared camera systems to work more productively by providing them with the most powerful

camera–software combination. Especially tailored software for predictive maintenance, R & D, and process monitoring is developed in-house. Most software is available in a wide variety of languages.

We support all our infrared cameras with a wide variety of accessories to adapt your equipment to the most demanding infrared applications.

14.2 Sharing our knowledge

Although our cameras are designed to be very user-friendly, there is a lot more to thermography than just knowing how to handle a camera. Therefore, FLIR Systems has founded the Infrared Training Center (ITC), a separate business unit, that provides certified training courses. Attending one of the ITC courses will give you a truly hands-on learning experience.

The staff of the ITC are also there to provide you with any application support you may need in putting infrared theory into practice.

14.3 Supporting our customers

FLIR Systems operates a worldwide service network to keep your camera running at all times. If you discover a problem with your camera, local service centers have all the equipment and expertise to solve it within the shortest possible time. Therefore, there is no need to send your camera to the other side of the world or to talk to someone who does not speak your language.

14.4 A few images from our facilities

10401303;a1

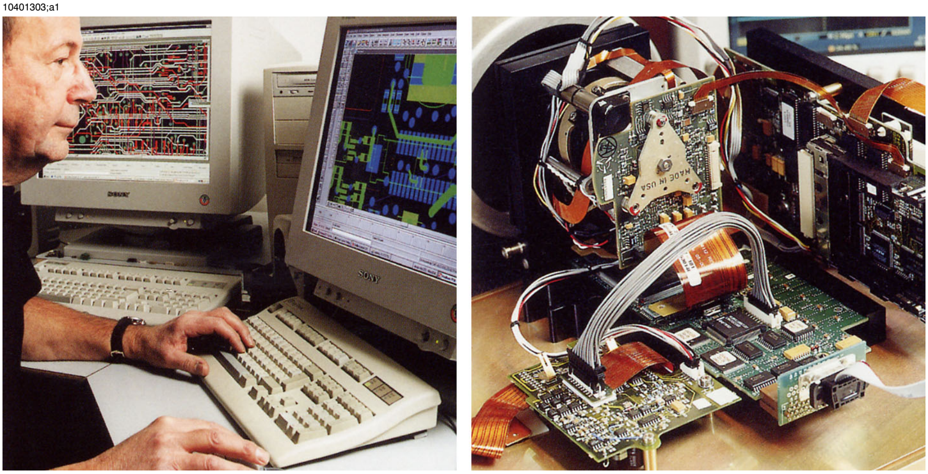


Figure 14.3 **LEFT:** Development of system electronics; **RIGHT:** Testing of an FPA detector

10401403;a1

Figure 14.4 **LEFT:** Diamond turning machine; **RIGHT:** Lens polishing

10401503;a1

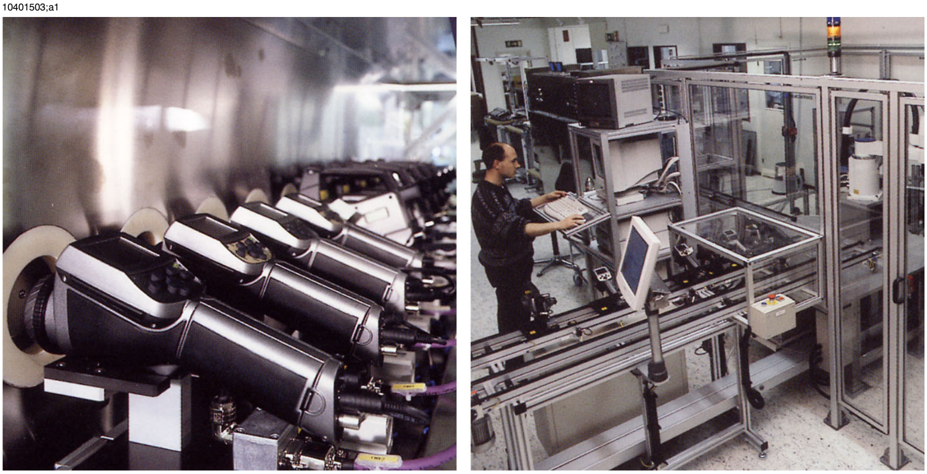


Figure 14.5 **LEFT:** Testing of infrared cameras in the climatic chamber; **RIGHT:** Robot used for camera testing and calibration

15 Thermographic measurement techniques

15.1 Introduction

An infrared camera measures and images the emitted infrared radiation from an object. The fact that radiation is a function of object surface temperature makes it possible for the camera to calculate and display this temperature.

However, the radiation measured by the camera does not only depend on the temperature of the object but is also a function of the emissivity. Radiation also originates from the surroundings and is reflected in the object. The radiation from the object and the reflected radiation will also be influenced by the absorption of the atmosphere.

To measure temperature accurately, it is therefore necessary to compensate for the effects of a number of different radiation sources. This is done on-line automatically by the camera. The following object parameters must, however, be supplied for the camera:

- The emissivity of the object
- The reflected apparent temperature
- The distance between the object and the camera
- The relative humidity
- Temperature of the atmosphere

15.2 Emissivity

The most important object parameter to set correctly is the emissivity which, in short, is a measure of how much radiation is emitted from the object, compared to that from a perfect blackbody of the same temperature.

Normally, object materials and surface treatments exhibit emissivity ranging from approximately 0.1 to 0.95. A highly polished (mirror) surface falls below 0.1, while an oxidized or painted surface has a higher emissivity. Oil-based paint, regardless of color in the visible spectrum, has an emissivity over 0.9 in the infrared. Human skin exhibits an emissivity 0.97 to 0.98.

Non-oxidized metals represent an extreme case of perfect opacity and high reflexivity, which does not vary greatly with wavelength. Consequently, the emissivity of metals is low – only increasing with temperature. For non-metals, emissivity tends to be high, and decreases with temperature.

15.2.1 Finding the emissivity of a sample

15.2.1.1 Step 1: Determining reflected apparent temperature

Use one of the following two methods to determine reflected apparent temperature:

15.2.1.1.1 Method 1: Direct method

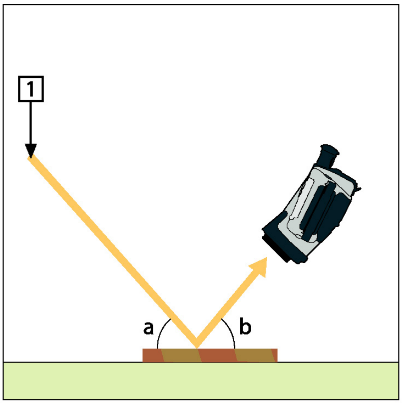
<table>
<tr><td>1</td><td>Look for possible reflection sources, considering that the incident angle = reflection angle (a = b).
10588903;a1


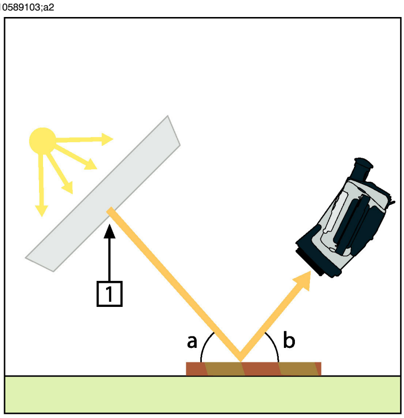
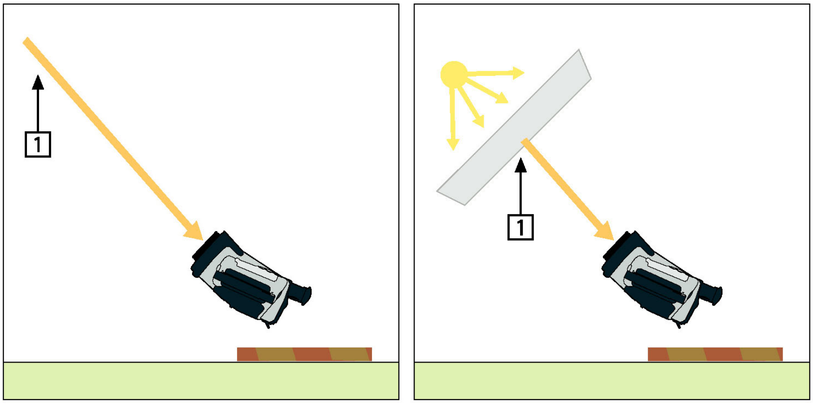
Figure 15.1 1 = Reflection source</td></tr>
<tr><td>2</td><td>If the reflection source is a spot source, modify the source by obstructing it using a piece if cardboard.
10589103;a2


Figure 15.2 1 = Reflection source</td></tr>
</table>

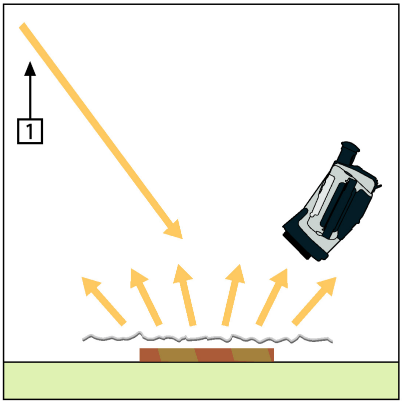
3	Measure the radiation intensity (= apparent temperature) from the reflecting source using the following settings: ■ Emissivity: 1.0 ■ D_{obj}: 0 You can measure the radiation intensity using one of the following two methods: 10589003;a2  **Figure 15.3** 1 = Reflection source

Note: Using a thermocouple to measure reflected apparent temperature is not recommended for two important reasons:

- A thermocouple does not measure radiation intensity
- A thermocouple requires a very good thermal contact to the surface, usually by gluing and covering the sensor by a thermal isolator.

15.2.1.1.2 Method 2: Reflector method

1	Crumble up a large piece of aluminum foil.
2	Uncrumble the aluminum foil and attach it to a piece of cardboard of the same size.
3	Put the piece of cardboard in front of the object you want to measure. Make sure that the side with aluminum foil points to the camera.
4	Set the emissivity to 1.0.

5	Measure the apparent temperature of the aluminum foil and write it down. 10727003;a2  **Figure 15.4** Measuring the apparent temperature of the aluminum foil

15.2.1.2 Step 2: Determining the emissivity

1	Select a place to put the sample.
2	Determine and set reflected apparent temperature according to the previous procedure.
3	Put a piece of electrical tape with known high emissivity on the sample.
4	Heat the sample at least 20 K above room temperature. Heating must be reasonably even.
5	Focus and auto-adjust the camera, and freeze the image.
6	Adjust **Level** and **Span** for best image brightness and contrast.
7	Set emissivity to that of the tape (usually 0.97).
8	Measure the temperature of the tape using one of the following measurement functions: ■ **Isotherm** (helps you to determine both the temperature and how evenly you have heated the sample) ■ **Spot** (simpler) ■ **Box Avg** (good for surfaces with varying emissivity).
9	Write down the temperature.
10	Move your measurement function to the sample surface.
11	Change the emissivity setting until you read the same temperature as your previous measurement.
12	Write down the emissivity.

Note:

- Avoid forced convection
- Look for a thermally stable surrounding that will not generate spot reflections
- Use high quality tape that you know is not transparent, and has a high emissivity you are certain of
- This method assumes that the temperature of your tape and the sample surface are the same. If they are not, your emissivity measurement will be wrong.

15.3 Reflected apparent temperature

This parameter is used to compensate for the radiation reflected in the object. If the emissivity is low and the object temperature relatively far from that of the reflected it will be important to set and compensate for the reflected apparent temperature correctly.

15.4 Distance

The distance is the distance between the object and the front lens of the camera. This parameter is used to compensate for the following two facts:

- That radiation from the target is absorbed by the athmosphere between the object and the camera.
- That radiation from the atmosphere itself is detected by the camera.

15.5 Relative humidity

The camera can also compensate for the fact that the transmittance is also dependent on the relative humidity of the atmosphere. To do this set the relative humidity to the correct value. For short distances and normal humidity the relative humidity can normally be left at a default value of 50%.

15.6 Other parameters

In addition, some cameras and analysis programs from FLIR Systems allow you to compensate for the following parameters:

- Atmospheric temperature – *i.e.* the temperature of the atmosphere between the camera and the target
- External optics temperature – *i.e.* the temperature of any external lenses or windows used in front of the camera
- External optics transmittance – *i.e.* the transmission of any external lenses or windows used in front of the camera

INTENTIONALLY LEFT BLANK

16 History of infrared technology

Before the year 1800, the existence of the infrared portion of the electromagnetic spectrum wasn't even suspected. The original significance of the infrared spectrum, or simply 'the infrared' as it is often called, as a form of heat radiation is perhaps less obvious today than it was at the time of its discovery by Herschel in 1800.

10398703;a1

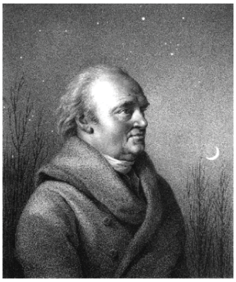


Figure 16.1 Sir William Herschel (1738–1822)

The discovery was made accidentally during the search for a new optical material. Sir William Herschel – Royal Astronomer to King George III of England, and already famous for his discovery of the planet Uranus – was searching for an optical filter material to reduce the brightness of the sun's image in telescopes during solar observations. While testing different samples of colored glass which gave similar reductions in brightness he was intrigued to find that some of the samples passed very little of the sun's heat, while others passed so much heat that he risked eye damage after only a few seconds' observation.

Herschel was soon convinced of the necessity of setting up a systematic experiment, with the objective of finding a single material that would give the desired reduction in brightness as well as the maximum reduction in heat. He began the experiment by actually repeating Newton's prism experiment, but looking for the heating effect rather than the visual distribution of intensity in the spectrum. He first blackened the bulb of a sensitive mercury-in-glass thermometer with ink, and with this as his radiation detector he proceeded to test the heating effect of the various colors of the spectrum formed on the top of a table by passing sunlight through a glass prism. Other thermometers, placed outside the sun's rays, served as controls.

As the blackened thermometer was moved slowly along the colors of the spectrum, the temperature readings showed a steady increase from the violet end to the red end. This was not entirely unexpected, since the Italian researcher, Landriani, in a similar experiment in 1777 had observed much the same effect. It was Herschel,

however, who was the first to recognize that there must be a point where the heating effect reaches a maximum, and that measurements confined to the visible portion of the spectrum failed to locate this point.

10398903;a1

Figure 16.2 Marsilio Landriani (1746–1815)

Moving the thermometer into the dark region beyond the red end of the spectrum, Herschel confirmed that the heating continued to increase. The maximum point, when he found it, lay well beyond the red end – in what is known today as the 'infrared wavelengths'.

When Herschel revealed his discovery, he referred to this new portion of the electromagnetic spectrum as the 'thermometrical spectrum'. The radiation itself he sometimes referred to as 'dark heat', or simply 'the invisible rays'. Ironically, and contrary to popular opinion, it wasn't Herschel who originated the term 'infrared'. The word only began to appear in print around 75 years later, and it is still unclear who should receive credit as the originator.

Herschel's use of glass in the prism of his original experiment led to some early controversies with his contemporaries about the actual existence of the infrared wavelengths. Different investigators, in attempting to confirm his work, used various types of glass indiscriminately, having different transparencies in the infrared. Through his later experiments, Herschel was aware of the limited transparency of glass to the newly-discovered thermal radiation, and he was forced to conclude that optics for the infrared would probably be doomed to the use of reflective elements exclusively (i.e. plane and curved mirrors). Fortunately, this proved to be true only until 1830, when the Italian investigator, Melloni, made his great discovery that naturally occurring rock salt (NaCl) – which was available in large enough natural crystals to be made into lenses and prisms – is remarkably transparent to the infrared. The result was that rock salt became the principal infrared optical material, and remained so for the next hundred years, until the art of synthetic crystal growing was mastered in the 1930's.

10399103;a1

Figure 16.3 Macedonio Melloni (1798–1854)

Thermometers, as radiation detectors, remained unchallenged until 1829, the year Nobili invented the thermocouple. (Herschel's own thermometer could be read to 0.2 °C (0.036 °F), and later models were able to be read to 0.05 °C (0.09 °F)). Then a breakthrough occurred; Melloni connected a number of thermocouples in series to form the first thermopile. The new device was at least 40 times as sensitive as the best thermometer of the day for detecting heat radiation – capable of detecting the heat from a person standing three meters away.

The first so-called 'heat-picture' became possible in 1840, the result of work by Sir John Herschel, son of the discoverer of the infrared and a famous astronomer in his own right. Based upon the differential evaporation of a thin film of oil when exposed to a heat pattern focused upon it, the thermal image could be seen by reflected light where the interference effects of the oil film made the image visible to the eye. Sir John also managed to obtain a primitive record of the thermal image on paper, which he called a 'thermograph'.

10399003;a2

Figure 16.4 Samuel P. Langley (1834–1906)

The improvement of infrared-detector sensitivity progressed slowly. Another major breakthrough, made by Langley in 1880, was the invention of the bolometer. This consisted of a thin blackened strip of platinum connected in one arm of a Wheatstone bridge circuit upon which the infrared radiation was focused and to which a sensitive galvanometer responded. This instrument is said to have been able to detect the heat from a cow at a distance of 400 meters.

An English scientist, Sir James Dewar, first introduced the use of liquefied gases as cooling agents (such as liquid nitrogen with a temperature of -196 °C (-320.8 °F)) in low temperature research. In 1892 he invented a unique vacuum insulating container in which it is possible to store liquefied gases for entire days. The common ‘thermos bottle’, used for storing hot and cold drinks, is based upon his invention.

Between the years 1900 and 1920, the inventors of the world ‘discovered’ the infrared. Many patents were issued for devices to detect personnel, artillery, aircraft, ships – and even icebergs. The first operating systems, in the modern sense, began to be developed during the 1914–18 war, when both sides had research programs devoted to the military exploitation of the infrared. These programs included experimental systems for enemy intrusion/detection, remote temperature sensing, secure communications, and ‘flying torpedo’ guidance. An infrared search system tested during this period was able to detect an approaching airplane at a distance of 1.5 km (0.94 miles), or a person more than 300 meters (984 ft.) away.

The most sensitive systems up to this time were all based upon variations of the bolometer idea, but the period between the two wars saw the development of two revolutionary new infrared detectors: the image converter and the photon detector. At first, the image converter received the greatest attention by the military, because it enabled an observer for the first time in history to literally ‘see in the dark’. However, the sensitivity of the image converter was limited to the near infrared wavelengths, and the most interesting military targets (i.e. enemy soldiers) had to be illuminated by infrared search beams. Since this involved the risk of giving away the observer’s position to a similarly-equipped enemy observer, it is understandable that military interest in the image converter eventually faded.

The tactical military disadvantages of so-called 'active’ (i.e. search beam-equipped) thermal imaging systems provided impetus following the 1939–45 war for extensive secret military infrared-research programs into the possibilities of developing ‘passive’ (no search beam) systems around the extremely sensitive photon detector. During this period, military secrecy regulations completely prevented disclosure of the status of infrared-imaging technology. This secrecy only began to be lifted in the middle of the 1950’s, and from that time adequate thermal-imaging devices finally began to be available to civilian science and industry.

17 Theory of thermography

17.1 Introduction

The subjects of infrared radiation and the related technique of thermography are still new to many who will use an infrared camera. In this section the theory behind thermography will be given.

17.2 The electromagnetic spectrum

The electromagnetic spectrum is divided arbitrarily into a number of wavelength regions, called *bands*, distinguished by the methods used to produce and detect the radiation. There is no fundamental difference between radiation in the different bands of the electromagnetic spectrum. They are all governed by the same laws and the only differences are those due to differences in wavelength.

10067803;a1

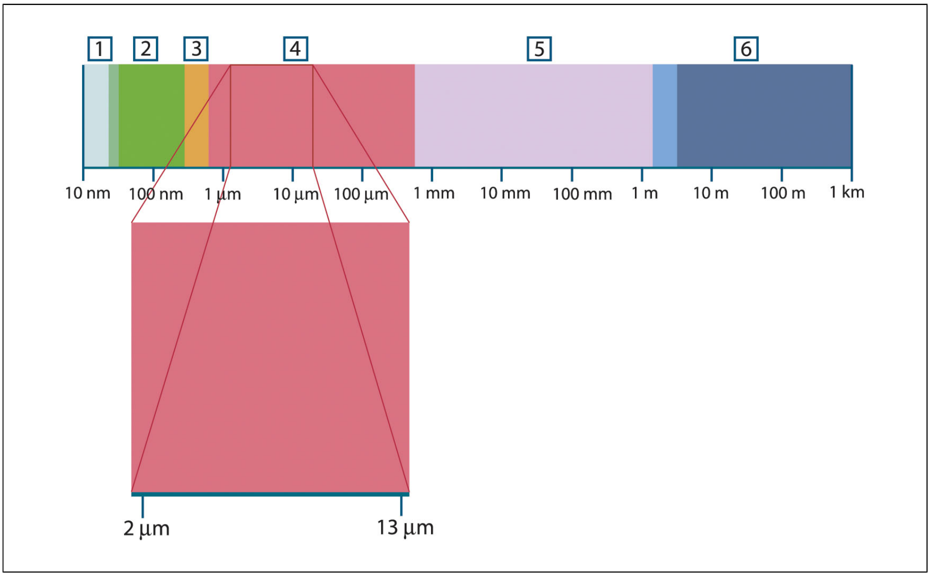


Figure 17.1 The electromagnetic spectrum. **1:** X-ray; **2:** UV; **3:** Visible; **4:** IR; **5:** Microwaves; **6:** Radiowaves.

Thermography makes use of the infrared spectral band. At the short-wavelength end the boundary lies at the limit of visual perception, in the deep red. At the long-wavelength end it merges with the microwave radio wavelengths, in the millimeter range.

The infrared band is often further subdivided into four smaller bands, the boundaries of which are also arbitrarily chosen. They include: the *near infrared* (0.75–3 µm), the *middle infrared* (3–6 µm), the *far infrared* (6–15 µm) and the *extreme infrared* (15–100

µm). Although the wavelengths are given in µm (micrometers), other units are often still used to measure wavelength in this spectral region, *e.g.* nanometer (nm) and Ångström (Å).

The relationships between the different wavelength measurements is:

$$10\ 000\ \text{Å} = 1\ 000\ \text{nm} = 1\ \mu = 1\ \mu\text{m}$$

17.3 Blackbody radiation

A blackbody is defined as an object which absorbs all radiation that impinges on it at any wavelength. The apparent misnomer *black* relating to an object emitting radiation is explained by Kirchhoff's Law (after *Gustav Robert Kirchhoff*, 1824–1887), which states that a body capable of absorbing all radiation at any wavelength is equally capable in the emission of radiation.

10398803;a1

Figure 17.2 Gustav Robert Kirchhoff (1824–1887)

The construction of a blackbody source is, in principle, very simple. The radiation characteristics of an aperture in an isotherm cavity made of an opaque absorbing material represents almost exactly the properties of a blackbody. A practical application of the principle to the construction of a perfect absorber of radiation consists of a box that is light tight except for an aperture in one of the sides. Any radiation which then enters the hole is scattered and absorbed by repeated reflections so only an infinitesimal fraction can possibly escape. The blackness which is obtained at the aperture is nearly equal to a blackbody and almost perfect for all wavelengths.

By providing such an isothermal cavity with a suitable heater it becomes what is termed a *cavity radiator*. An isothermal cavity heated to a uniform temperature generates blackbody radiation, the characteristics of which are determined solely by the temperature of the cavity. Such cavity radiators are commonly used as sources of radiation in temperature reference standards in the laboratory for calibrating thermographic instruments, such as a FLIR Systems camera for example.

If the temperature of blackbody radiation increases to more than 525°C (977°F), the source begins to be visible so that it appears to the eye no longer black. This is the incipient red heat temperature of the radiator, which then becomes orange or yellow as the temperature increases further. In fact, the definition of the so-called *color temperature* of an object is the temperature to which a blackbody would have to be heated to have the same appearance.

Now consider three expressions that describe the radiation emitted from a blackbody.

17.3.1 Planck's law

10399203;a1

Figure 17.3 Max Planck (1858–1947)

Max Planck (1858–1947) was able to describe the spectral distribution of the radiation from a blackbody by means of the following formula:

$$W_{\lambda b} = \frac{2\pi hc^2}{\lambda^5 \left(e^{hc/\lambda kT} - 1\right)} \times 10^{-6} [Watt / m^2, \mu m]$$

where:

$W_{\lambda b}$	Blackbody spectral radiant emittance at wavelength λ.
c	Velocity of light = 3×10^8 m/s
h	Planck's constant = 6.6×10^{-34} Joule sec.
k	Boltzmann's constant = 1.4×10^{-23} Joule/K.
T	Absolute temperature (K) of a blackbody.
λ	Wavelength (μm).

➲ The factor 10^{-6} is used since spectral emittance in the curves is expressed in Watt/m^2, μm.

Planck's formula, when plotted graphically for various temperatures, produces a family of curves. Following any particular Planck curve, the spectral emittance is zero at $\lambda = 0$, then increases rapidly to a maximum at a wavelength λ_{max} and after passing it approaches zero again at very long wavelengths. The higher the temperature, the shorter the wavelength at which maximum occurs.

10327103;a4

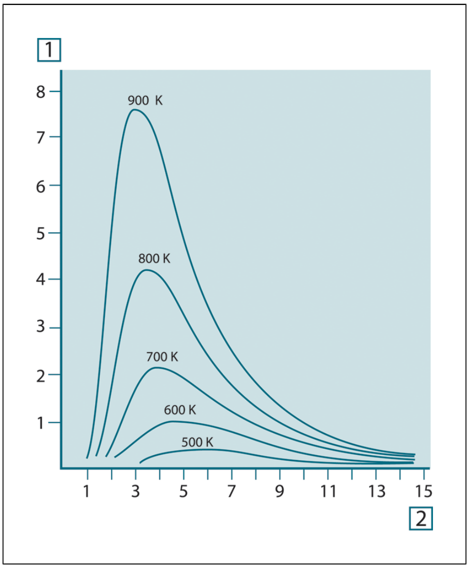


Figure 17.4 Blackbody spectral radiant emittance according to Planck's law, plotted for various absolute temperatures. **1:** Spectral radiant emittance (W/cm^2 × 10^3(μm)); **2:** Wavelength (μm)

17.3.2 Wien's displacement law

By differentiating Planck's formula with respect to λ, and finding the maximum, we have:

$$\lambda_{\max} = \frac{2898}{T}[\mu m]$$

This is Wien's formula (after *Wilhelm Wien*, 1864–1928), which expresses mathematically the common observation that colors vary from red to orange or yellow as the temperature of a thermal radiator increases. The wavelength of the color is the same as the wavelength calculated for λ_{max}. A good approximation of the value of λ_{max} for a given blackbody temperature is obtained by applying the rule-of-thumb 3 000/T

µm. Thus, a very hot star such as Sirius (11 000 K), emitting bluish-white light, radiates with the peak of spectral radiant emittance occurring within the invisible ultraviolet spectrum, at wavelength 0.27 µm.

10399403;a1

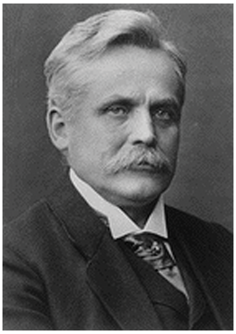


Figure 17.5 Wilhelm Wien (1864–1928)

The sun (approx. 6 000 K) emits yellow light, peaking at about 0.5 µm in the middle of the visible light spectrum.

At room temperature (300 K) the peak of radiant emittance lies at 9.7 µm, in the far infrared, while at the temperature of liquid nitrogen (77 K) the maximum of the almost insignificant amount of radiant emittance occurs at 38 µm, in the extreme infrared wavelengths.

10327203;a4

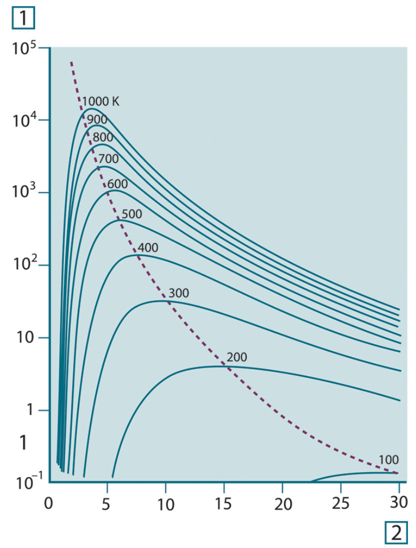


Figure 17.6 Planckian curves plotted on semi-log scales from 100 K to 1000 K. The dotted line represents the locus of maximum radiant emittance at each temperature as described by Wien's displacement law. **1:** Spectral radiant emittance (W/cm^2 (µm)); **2:** Wavelength (µm).

17.3.3 Stefan-Boltzmann's law

By integrating Planck's formula from λ = 0 to λ = ∞, we obtain the total radiant emittance (W_b) of a blackbody:

$$W_b = \sigma T^4 \ [\mathrm{Watt/m^2}]$$

This is the Stefan-Boltzmann formula (after *Josef Stefan*, 1835–1893, and *Ludwig Boltzmann*, 1844–1906), which states that the total emissive power of a blackbody is proportional to the fourth power of its absolute temperature. Graphically, W_b represents the area below the Planck curve for a particular temperature. It can be shown that the radiant emittance in the interval λ = 0 to λ_{max} is only 25% of the total, which represents about the amount of the sun's radiation which lies inside the visible light spectrum.

10399303;a1

Figure 17.7 Josef Stefan (1835–1893), and Ludwig Boltzmann (1844–1906)

Using the Stefan-Boltzmann formula to calculate the power radiated by the human body, at a temperature of 300 K and an external surface area of approx. 2 m^2, we obtain 1 kW. This power loss could not be sustained if it were not for the compensating absorption of radiation from surrounding surfaces, at room temperatures which do not vary too drastically from the temperature of the body – or, of course, the addition of clothing.

17.3.4 Non-blackbody emitters

So far, only blackbody radiators and blackbody radiation have been discussed. However, real objects almost never comply with these laws over an extended wavelength region – although they may approach the blackbody behavior in certain spectral intervals. For example, a certain type of white paint may appear perfectly *white* in the visible light spectrum, but becomes distinctly *gray* at about 2 µm, and beyond 3 µm it is almost *black*.

There are three processes which can occur that prevent a real object from acting like a blackbody: a fraction of the incident radiation α may be absorbed, a fraction ρ may be reflected, and a fraction τ may be transmitted. Since all of these factors are more or less wavelength dependent, the subscript λ is used to imply the spectral dependence of their definitions. Thus:

- The spectral absorptance α_λ= the ratio of the spectral radiant power absorbed by an object to that incident upon it.
- The spectral reflectance ρ_λ = the ratio of the spectral radiant power reflected by an object to that incident upon it.
- The spectral transmittance τ_λ = the ratio of the spectral radiant power transmitted through an object to that incident upon it.

The sum of these three factors must always add up to the whole at any wavelength, so we have the relation:

$$\alpha_\lambda + \rho_\lambda + \tau_\lambda = 1$$

For opaque materials τ_λ = 0 and the relation simplifies to:

$$\alpha_\lambda + \rho_\lambda = 1$$

Another factor, called the emissivity, is required to describe the fraction ε of the radiant emittance of a blackbody produced by an object at a specific temperature. Thus, we have the definition:

The spectral emissivity ε_λ= the ratio of the spectral radiant power from an object to that from a blackbody at the same temperature and wavelength.

Expressed mathematically, this can be written as the ratio of the spectral emittance of the object to that of a blackbody as follows:

$$\varepsilon_\lambda = \frac{W_{\lambda o}}{W_{\lambda b}}$$

Generally speaking, there are three types of radiation source, distinguished by the ways in which the spectral emittance of each varies with wavelength.

- A blackbody, for which ε_λ = ε = 1
- A graybody, for which ε_λ = ε = constant less than 1
- A selective radiator, for which ε varies with wavelength

According to Kirchhoff's law, for any material the spectral emissivity and spectral absorptance of a body are equal at any specified temperature and wavelength. That is:

$$\varepsilon_\lambda = \alpha_\lambda$$

From this we obtain, for an opaque material (since α_λ + ρ_λ = 1):

$$\varepsilon_\lambda + \rho_\lambda = 1$$

For highly polished materials ε_λ approaches zero, so that for a perfectly reflecting material (*i.e.* a perfect mirror) we have:

$$\rho_\lambda = 1$$

For a graybody radiator, the Stefan-Boltzmann formula becomes:

$$W = \varepsilon \sigma T^4 \ [\mathrm{Watt/m^2}]$$

This states that the total emissive power of a graybody is the same as a blackbody at the same temperature reduced in proportion to the value of ε from the graybody.

10401203;a2

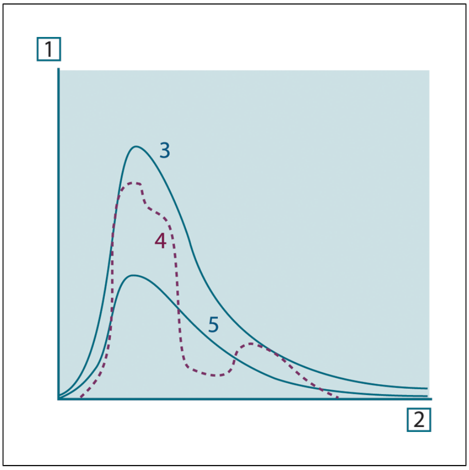


Figure 17.8 Spectral radiant emittance of three types of radiators. **1:** Spectral radiant emittance; **2:** Wavelength; **3:** Blackbody; **4:** Selective radiator; **5:** Graybody.

10327303;a4

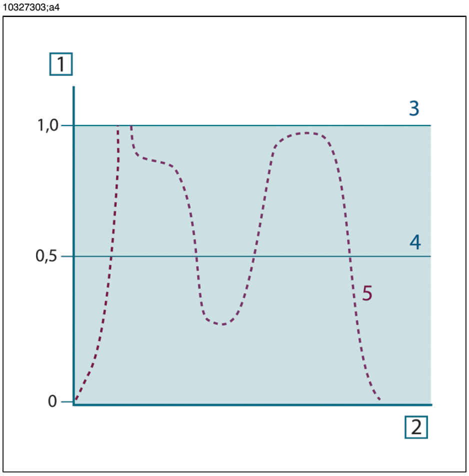


Figure 17.9 Spectral emissivity of three types of radiators. **1:** Spectral emissivity; **2:** Wavelength; **3:** Blackbody; **4:** Graybody; **5:** Selective radiator.

17.4 Infrared semi-transparent materials

Consider now a non-metallic, semi-transparent body – let us say, in the form of a thick flat plate of plastic material. When the plate is heated, radiation generated within its volume must work its way toward the surfaces through the material in which it is partially absorbed. Moreover, when it arrives at the surface, some of it is reflected back into the interior. The back-reflected radiation is again partially absorbed, but

some of it arrives at the other surface, through which most of it escapes; part of it is reflected back again. Although the progressive reflections become weaker and weaker they must all be added up when the total emittance of the plate is sought. When the resulting geometrical series is summed, the effective emissivity of a semi-transparent plate is obtained as:

$$\varepsilon_\lambda = \frac{(1-\rho_\lambda)(1-\tau_\lambda)}{1-\rho_\lambda\tau_\lambda}$$

When the plate becomes opaque this formula is reduced to the single formula:

$$\varepsilon_\lambda = 1-\rho_\lambda$$

This last relation is a particularly convenient one, because it is often easier to measure reflectance than to measure emissivity directly.

18 The measurement formula

As already mentioned, when viewing an object, the camera receives radiation not only from the object itself. It also collects radiation from the surroundings reflected via the object surface. Both these radiation contributions become attenuated to some extent by the atmosphere in the measurement path. To this comes a third radiation contribution from the atmosphere itself.

This description of the measurement situation, as illustrated in the figure below, is so far a fairly true description of the real conditions. What has been neglected could for instance be sun light scattering in the atmosphere or stray radiation from intense radiation sources outside the field of view. Such disturbances are difficult to quantify, however, in most cases they are fortunately small enough to be neglected. In case they are not negligible, the measurement configuration is likely to be such that the risk for disturbance is obvious, at least to a trained operator. It is then his responsibility to modify the measurement situation to avoid the disturbance e.g. by changing the viewing direction, shielding off intense radiation sources etc.

Accepting the description above, we can use the figure below to derive a formula for the calculation of the object temperature from the calibrated camera output.

10400503;a1

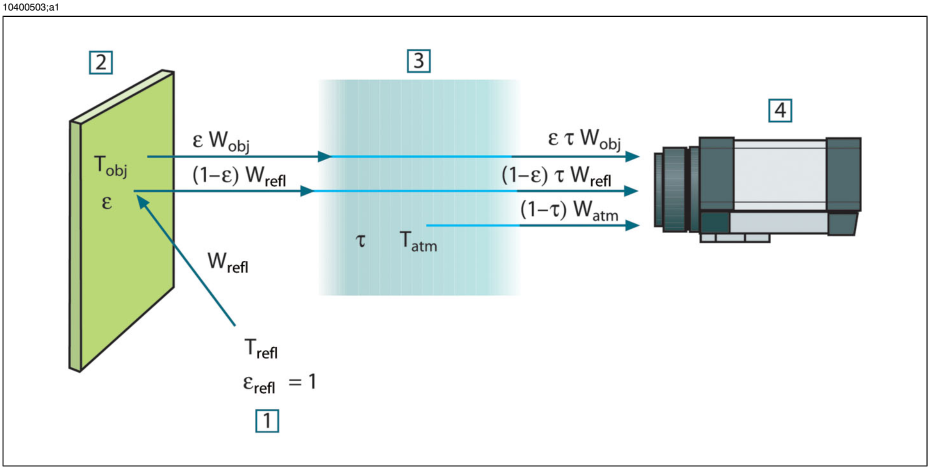


Figure 18.1 A schematic representation of the general thermographic measurement situation.**1:** Surroundings; **2:** Object; **3:** Atmosphere; **4:** Camera

Assume that the received radiation power W from a blackbody source of temperature T_{source} on short distance generates a camera output signal U_{source} that is proportional to the power input (power linear camera). We can then write (Equation 1):

$$U_{source} = CW(T_{source})$$

or, with simplified notation:

$$U_{source} = CW_{source}$$

where C is a constant.

Should the source be a graybody with emittance ε, the received radiation would consequently be εW_{source}.

We are now ready to write the three collected radiation power terms:

1 – Emission from the object = $\varepsilon\tau W_{obj}$, where ε is the emittance of the object and τ is the transmittance of the atmosphere. The object temperature is T_{obj}.

2 – Reflected emission from ambient sources = $(1 – \varepsilon)\tau W_{refl}$, where $(1 – \varepsilon)$ is the reflectance of the object. The ambient sources have the temperature T_{refl}.

It has here been assumed that the temperature T_{refl} is the same for all emitting surfaces within the halfsphere seen from a point on the object surface. This is of course sometimes a simplification of the true situation. It is, however, a necessary simplification in order to derive a workable formula, and T_{refl} can – at least theoretically – be given a value that represents an efficient temperature of a complex surrounding.

Note also that we have assumed that the emittance for the surroundings = 1. This is correct in accordance with Kirchhoff's law: All radiation impinging on the surrounding surfaces will eventually be absorbed by the same surfaces. Thus the emittance = 1. (Note though that the latest discussion requires the complete sphere around the object to be considered.)

3 – Emission from the atmosphere = $(1 – \tau)\tau W_{atm}$, where $(1 – \tau)$ is the emittance of the atmosphere. The temperature of the atmosphere is T_{atm}.

The total received radiation power can now be written (Equation 2):

$$W_{tot} = \varepsilon\tau W_{obj} + (1-\varepsilon)\tau W_{refl} + (1-\tau)W_{atm}$$

We multiply each term by the constant C of Equation 1 and replace the CW products by the corresponding U according to the same equation, and get (Equation 3):

$$U_{tot} = \varepsilon\tau U_{obj} + (1-\varepsilon)\tau U_{refl} + (1-\tau)U_{atm}$$

Solve Equation 3 for U_{obj} (Equation 4):

$$U_{obj} = \frac{1}{\varepsilon\tau}U_{tot} - \frac{1-\varepsilon}{\varepsilon}U_{refl} - \frac{1-\tau}{\varepsilon\tau}U_{atm}$$

This is the general measurement formula used in all the FLIR Systems thermographic equipment. The voltages of the formula are:

Figure 18.2 Voltages

U_{obj}	Calculated camera output voltage for a blackbody of temperature T_{obj} i.e. a voltage that can be directly converted into true requested object temperature.
U_{tot}	Measured camera output voltage for the actual case.
U_{refl}	Theoretical camera output voltage for a blackbody of temperature T_{refl} according to the calibration.
U_{atm}	Theoretical camera output voltage for a blackbody of temperature T_{atm} according to the calibration.

The operator has to supply a number of parameter values for the calculation:

- the object emittance ε,
- the relative humidity,
- T_{atm}
- object distance (D_{obj})
- the (effective) temperature of the object surroundings, or the reflected ambient temperature T_{refl}, and
- the temperature of the atmosphere T_{atm}

This task could sometimes be a heavy burden for the operator since there are normally no easy ways to find accurate values of emittance and atmospheric transmittance for the actual case. The two temperatures are normally less of a problem provided the surroundings do not contain large and intense radiation sources.

A natural question in this connection is: How important is it to know the right values of these parameters? It could though be of interest to get a feeling for this problem already here by looking into some different measurement cases and compare the relative magnitudes of the three radiation terms. This will give indications about when it is important to use correct values of which parameters.

The figures below illustrates the relative magnitudes of the three radiation contributions for three different object temperatures, two emittances, and two spectral ranges: SW and LW. Remaining parameters have the following fixed values:

- τ = 0.88
- T_{refl} = +20°C (+68°F)
- T_{atm} = +20°C (+68°F)

It is obvious that measurement of low object temperatures are more critical than measuring high temperatures since the 'disturbing' radiation sources are relatively much stronger in the first case. Should also the object emittance be low, the situation would be still more difficult.

We have finally to answer a question about the importance of being allowed to use the calibration curve above the highest calibration point, what we call extrapolation. Imagine that we in a certain case measure $U_{tot} = 4.5$ volts. The highest calibration point for the camera was in the order of 4.1 volts, a value unknown to the operator. Thus, even if the object happened to be a blackbody, i.e. $U_{obj} = U_{tot}$, we are actually performing extrapolation of the calibration curve when converting 4.5 volts into temperature.

Let us now assume that the object is not black, it has an emittance of 0.75, and the transmittance is 0.92. We also assume that the two second terms of Equation 4 amount to 0.5 volts together. Computation of U_{obj} by means of Equation 4 then results in $U_{obj} = 4.5 / 0.75 / 0.92 – 0.5 = 6.0$. This is a rather extreme extrapolation, particularly when considering that the video amplifier might limit the output to 5 volts! Note, though, that the application of the calibration curve is a theoretical procedure where no electronic or other limitations exist. We trust that if there had been no signal limitations in the camera, and if it had been calibrated far beyond 5 volts, the resulting curve would have been very much the same as our real curve extrapolated beyond 4.1 volts, provided the calibration algorithm is based on radiation physics, like the FLIR Systems algorithm. Of course there must be a limit to such extrapolations.

10400603;a2

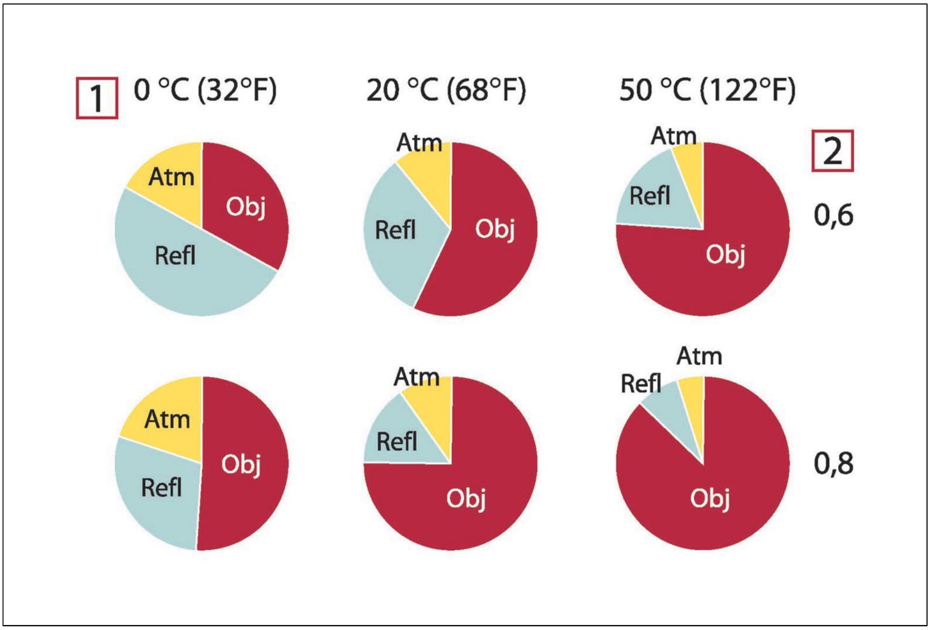


Figure 18.3 Relative magnitudes of radiation sources under varying measurement conditions (SW camera). **1:** Object temperature; **2:** Emittance; **Obj:** Object radiation; **Refl:** Reflected radiation; **Atm:** atmosphere radiation. Fixed parameters: $\tau = 0.88$; $T_{refl} = 20°C (+68°F)$; $T_{atm} = 20°C (+68°F)$.

18

10400703;a2

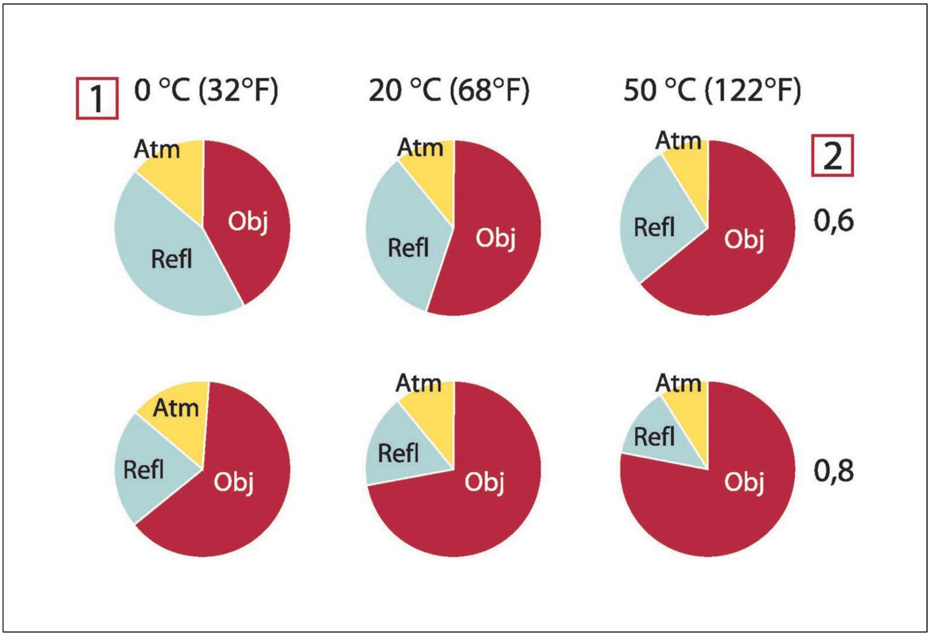


Figure 18.4 Relative magnitudes of radiation sources under varying measurement conditions (LW camera). **1:** Object temperature; **2:** Emittance; **Obj:** Object radiation; **Refl:** Reflected radiation; **Atm:** atmosphere radiation. Fixed parameters: $\tau = 0.88$; $T_{refl} = 20°C$ (+68°F); $T_{atm} = 20°C$ (+68°F).

19 Emissivity tables

This section presents a compilation of emissivity data from the infrared literature and measurements made by FLIR Systems.

19.1 References

1	Mikaél A. Bramson: *Infrared Radiation, A Handbook for Applications*, Plenum press, N.Y.
2	William L. Wolfe, George J. Zissis: *The Infrared Handbook*, Office of Naval Research, Department of Navy, Washington, D.C.
3	Madding, R. P.: *Thermographic Instruments and systems*. Madison, Wisconsin: University of Wisconsin – Extension, Department of Engineering and Applied Science.
4	William L. Wolfe: *Handbook of Military Infrared Technology*, Office of Naval Research, Department of Navy, Washington, D.C.
5	Jones, Smith, Probert: *External thermography of buildings...,* Proc. of the Society of Photo-Optical Instrumentation Engineers, vol.110, Industrial and Civil Applications of Infrared Technology, June 1977 London.
6	Paljak, Pettersson: *Thermography of Buildings*, Swedish Building Research Institute, Stockholm 1972.
7	Vlcek, J: *Determination of emissivity with imaging radiometers and some emissivities at* $\lambda = 5\ \mu m$*.* Photogrammetric Engineering and Remote Sensing.
8	Kern: *Evaluation of infrared emission of clouds and ground as measured by weather satellites*, Defence Documentation Center, AD 617 417.
9	Öhman, Claes: *Emittansmätningar med AGEMA E-Box*. Teknisk rapport, AGEMA 1999. (Emittance measurements using AGEMA E-Box. Technical report, AGEMA 1999.)
10	Matteï, S., Tang-Kwor, E: *Emissivity measurements for Nextel Velvet coating 811-21 between –36°C AND 82°C.*
11	Lohrengel & Todtenhaupt (1996)
12	ITC Technical publication 32.
13	ITC Technical publication 29.

19.2 Important note about the emissivity tables

The type of camera that has been used when compiling the emissivity data is specified in column 4. The values should be regarded as recommendations only and used with caution.

19.3 Tables

Figure 19.1 **1:** Material; **2:** Specification; **3:** Temperature in °C; **4:** Spectrum (**T:** Total spectrum; **SW:** 2–5 μm; **LW:** 8–14 μm, **LLW:** 6.5–20 μm); **5:** Emissivity: **6:** Reference to literature source above

1	2	3	4	5	6
3M type 35	Vinyl electrical tape (several colors)	< 80	LW	Ca. 0.96	13
3M type 88	Black vinyl electrical tape	< 105	LW	Ca. 0.96	13
3M type 88	Black vinyl electrical tape	< 105	MW	< 0.96	13
3M type Super 33+	Black vinyl electrical tape	< 80	LW	Ca. 0.96	13
Aluminum	anodized, black, dull	70	LW	0.95	9
Aluminum	anodized, black, dull	70	SW	0.67	9
Aluminum	anodized, light gray, dull	70	LW	0.97	9
Aluminum	anodized, light gray, dull	70	SW	0.61	9
Aluminum	anodized sheet	100	T	0.55	2
Aluminum	as received, plate	100	T	0.09	4
Aluminum	as received, sheet	100	T	0.09	2
Aluminum	cast, blast cleaned	70	LW	0.46	9
Aluminum	cast, blast cleaned	70	SW	0.47	9
Aluminum	dipped in HNO_3, plate	100	T	0.05	4
Aluminum	foil	27	3 μm	0.09	3
Aluminum	foil	27	10 μm	0.04	3
Aluminum	oxidized, strongly	50–500	T	0.2–0.3	1
Aluminum	polished	50–100	T	0.04–0.06	1
Aluminum	polished, sheet	100	T	0.05	2
Aluminum	polished plate	100	T	0.05	4

1	2	3	4	5	6
Aluminum	roughened	27	3 μm	0.28	3
Aluminum	roughened	27	10 μm	0.18	3
Aluminum	rough surface	20–50	T	0.06–0.07	1
Aluminum	sheet, 4 samples differently scratched	70	LW	0.03–0.06	9
Aluminum	sheet, 4 samples differently scratched	70	SW	0.05–0.08	9
Aluminum	vacuum deposited	20	T	0.04	2
Aluminum	weathered, heavily	17	SW	0.83–0.94	5
Aluminum bronze		20	T	0.60	1
Aluminum hydroxide	powder		T	0.28	1
Aluminum oxide	activated, powder		T	0.46	1
Aluminum oxide	pure, powder (alumina)		T	0.16	1
Asbestos	board	20	T	0.96	1
Asbestos	fabric		T	0.78	1
Asbestos	floor tile	35	SW	0.94	7
Asbestos	paper	40–400	T	0.93–0.95	1
Asbestos	powder		T	0.40–0.60	1
Asbestos	slate	20	T	0.96	1
Asphalt paving		4	LLW	0.967	8
Brass	dull, tarnished	20–350	T	0.22	1
Brass	oxidized	70	SW	0.04–0.09	9
Brass	oxidized	70	LW	0.03–0.07	9
Brass	oxidized	100	T	0.61	2
Brass	oxidized at 600°C	200–600	T	0.59–0.61	1
Brass	polished	200	T	0.03	1
Brass	polished, highly	100	T	0.03	2

19

1	2	3	4	5	6
Brass	rubbed with 80-grit emery	20	T	0.20	2
Brass	sheet, rolled	20	T	0.06	1
Brass	sheet, worked with emery	20	T	0.2	1
Brick	alumina	17	SW	0.68	5
Brick	common	17	SW	0.86–0.81	5
Brick	Dinas silica, glazed, rough	1100	T	0.85	1
Brick	Dinas silica, refractory	1000	T	0.66	1
Brick	Dinas silica, unglazed, rough	1000	T	0.80	1
Brick	firebrick	17	SW	0.68	5
Brick	fireclay	20	T	0.85	1
Brick	fireclay	1000	T	0.75	1
Brick	fireclay	1200	T	0.59	1
Brick	masonry	35	SW	0.94	7
Brick	masonry, plastered	20	T	0.94	1
Brick	red, common	20	T	0.93	2
Brick	red, rough	20	T	0.88–0.93	1
Brick	refractory, corundum	1000	T	0.46	1
Brick	refractory, magnesite	1000–1300	T	0.38	1
Brick	refractory, strongly radiating	500–1000	T	0.8–0.9	1
Brick	refractory, weakly radiating	500–1000	T	0.65–0.75	1
Brick	silica, 95% SiO_2	1230	T	0.66	1
Brick	sillimanite, 33% SiO_2, 64% Al_2O_3	1500	T	0.29	1

1	2	3	4	5	6
Brick	waterproof	17	SW	0.87	5
Bronze	phosphor bronze	70	LW	0.06	9
Bronze	phosphor bronze	70	SW	0.08	9
Bronze	polished	50	T	0.1	1
Bronze	porous, rough	50–150	T	0.55	1
Bronze	powder		T	0.76–0.80	1
Carbon	candle soot	20	T	0.95	2
Carbon	charcoal powder		T	0.96	1
Carbon	graphite, filed surface	20	T	0.98	2
Carbon	graphite powder		T	0.97	1
Carbon	lampblack	20–400	T	0.95–0.97	1
Chipboard	untreated	20	SW	0.90	6
Chromium	polished	50	T	0.10	1
Chromium	polished	500–1000	T	0.28–0.38	1
Clay	fired	70	T	0.91	1
Cloth	black	20	T	0.98	1
Concrete		20	T	0.92	2
Concrete	dry	36	SW	0.95	7
Concrete	rough	17	SW	0.97	5
Concrete	walkway	5	LLW	0.974	8
Copper	commercial, burnished	20	T	0.07	1
Copper	electrolytic, carefully polished	80	T	0.018	1
Copper	electrolytic, polished	–34	T	0.006	4
Copper	molten	1100–1300	T	0.13–0.15	1
Copper	oxidized	50	T	0.6–0.7	1
Copper	oxidized, black	27	T	0.78	4

19

1	2	3	4	5	6
Copper	oxidized, heavily	20	T	0.78	2
Copper	oxidized to blackness		T	0.88	1
Copper	polished	50–100	T	0.02	1
Copper	polished	100	T	0.03	2
Copper	polished, commercial	27	T	0.03	4
Copper	polished, mechanical	22	T	0.015	4
Copper	pure, carefully prepared surface	22	T	0.008	4
Copper	scraped	27	T	0.07	4
Copper dioxide	powder		T	0.84	1
Copper oxide	red, powder		T	0.70	1
Ebonite			T	0.89	1
Emery	coarse	80	T	0.85	1
Enamel		20	T	0.9	1
Enamel	lacquer	20	T	0.85–0.95	1
Fiber board	hard, untreated	20	SW	0.85	6
Fiber board	masonite	70	LW	0.88	9
Fiber board	masonite	70	SW	0.75	9
Fiber board	particle board	70	LW	0.89	9
Fiber board	particle board	70	SW	0.77	9
Fiber board	porous, untreated	20	SW	0.85	6
Gold	polished	130	T	0.018	1
Gold	polished, carefully	200–600	T	0.02–0.03	1
Gold	polished, highly	100	T	0.02	2
Granite	polished	20	LLW	0.849	8
Granite	rough	21	LLW	0.879	8
Granite	rough, 4 different samples	70	LW	0.77–0.87	9

Publ. No. T559015 Rev. a590 – ENGLISH (EN) – February 29, 2012

1	2	3	4	5	6
Granite	rough, 4 different samples	70	SW	0.95–0.97	9
Gypsum		20	T	0.8–0.9	1
Ice: See Water					
Iron, cast	casting	50	T	0.81	1
Iron, cast	ingots	1000	T	0.95	1
Iron, cast	liquid	1300	T	0.28	1
Iron, cast	machined	800–1000	T	0.60–0.70	1
Iron, cast	oxidized	38	T	0.63	4
Iron, cast	oxidized	100	T	0.64	2
Iron, cast	oxidized	260	T	0.66	4
Iron, cast	oxidized	538	T	0.76	4
Iron, cast	oxidized at 600°C	200–600	T	0.64–0.78	1
Iron, cast	polished	38	T	0.21	4
Iron, cast	polished	40	T	0.21	2
Iron, cast	polished	200	T	0.21	1
Iron, cast	unworked	900–1100	T	0.87–0.95	1
Iron and steel	cold rolled	70	LW	0.09	9
Iron and steel	cold rolled	70	SW	0.20	9
Iron and steel	covered with red rust	20	T	0.61–0.85	1
Iron and steel	electrolytic	22	T	0.05	4
Iron and steel	electrolytic	100	T	0.05	4
Iron and steel	electrolytic	260	T	0.07	4
Iron and steel	electrolytic, carefully polished	175–225	T	0.05–0.06	1
Iron and steel	freshly worked with emery	20	T	0.24	1
Iron and steel	ground sheet	950–1100	T	0.55–0.61	1
Iron and steel	heavily rusted sheet	20	T	0.69	2

19

1	2	3	4	5	6
Iron and steel	hot rolled	20	T	0.77	1
Iron and steel	hot rolled	130	T	0.60	1
Iron and steel	oxidized	100	T	0.74	1
Iron and steel	oxidized	100	T	0.74	4
Iron and steel	oxidized	125–525	T	0.78–0.82	1
Iron and steel	oxidized	200	T	0.79	2
Iron and steel	oxidized	1227	T	0.89	4
Iron and steel	oxidized	200–600	T	0.80	1
Iron and steel	oxidized strongly	50	T	0.88	1
Iron and steel	oxidized strongly	500	T	0.98	1
Iron and steel	polished	100	T	0.07	2
Iron and steel	polished	400–1000	T	0.14–0.38	1
Iron and steel	polished sheet	750–1050	T	0.52–0.56	1
Iron and steel	rolled, freshly	20	T	0.24	1
Iron and steel	rolled sheet	50	T	0.56	1
Iron and steel	rough, plane surface	50	T	0.95–0.98	1
Iron and steel	rusted, heavily	17	SW	0.96	5
Iron and steel	rusted red, sheet	22	T	0.69	4
Iron and steel	rusty, red	20	T	0.69	1
Iron and steel	shiny, etched	150	T	0.16	1
Iron and steel	shiny oxide layer, sheet,	20	T	0.82	1
Iron and steel	wrought, carefully polished	40–250	T	0.28	1
Iron galvanized	heavily oxidized	70	LW	0.85	9
Iron galvanized	heavily oxidized	70	SW	0.64	9
Iron galvanized	sheet	92	T	0.07	4
Iron galvanized	sheet, burnished	30	T	0.23	1
Iron galvanized	sheet, oxidized	20	T	0.28	1

1	2	3	4	5	6
Iron tinned	sheet	24	T	0.064	4
Krylon Ultra-flat black 1602	Flat black	Room temperature up to 175	LW	Ca. 0.96	12
Krylon Ultra-flat black 1602	Flat black	Room temperature up to 175	MW	Ca. 0.97	12
Lacquer	3 colors sprayed on Aluminum	70	LW	0.92–0.94	9
Lacquer	3 colors sprayed on Aluminum	70	SW	0.50–0.53	9
Lacquer	Aluminum on rough surface	20	T	0.4	1
Lacquer	bakelite	80	T	0.83	1
Lacquer	black, dull	40–100	T	0.96–0.98	1
Lacquer	black, matte	100	T	0.97	2
Lacquer	black, shiny, sprayed on iron	20	T	0.87	1
Lacquer	heat–resistant	100	T	0.92	1
Lacquer	white	40–100	T	0.8–0.95	1
Lacquer	white	100	T	0.92	2
Lead	oxidized, gray	20	T	0.28	1
Lead	oxidized, gray	22	T	0.28	4
Lead	oxidized at 200°C	200	T	0.63	1
Lead	shiny	250	T	0.08	1
Lead	unoxidized, polished	100	T	0.05	4
Lead red		100	T	0.93	4
Lead red, powder		100	T	0.93	1
Leather	tanned		T	0.75–0.80	1
Lime			T	0.3–0.4	1
Magnesium		22	T	0.07	4
Magnesium		260	T	0.13	4

1	2	3	4	5	6
Magnesium		538	T	0.18	4
Magnesium	polished	20	T	0.07	2
Magnesium powder			T	0.86	1
Molybdenum		600–1000	T	0.08–0.13	1
Molybdenum		1500–2200	T	0.19–0.26	1
Molybdenum	filament	700–2500	T	0.1–0.3	1
Mortar		17	SW	0.87	5
Mortar	dry	36	SW	0.94	7
Nextel Velvet 811-21 Black	Flat black	–60–150	LW	> 0.97	10 and 11
Nichrome	rolled	700	T	0.25	1
Nichrome	sandblasted	700	T	0.70	1
Nichrome	wire, clean	50	T	0.65	1
Nichrome	wire, clean	500–1000	T	0.71–0.79	1
Nichrome	wire, oxidized	50–500	T	0.95–0.98	1
Nickel	bright matte	122	T	0.041	4
Nickel	commercially pure, polished	100	T	0.045	1
Nickel	commercially pure, polished	200–400	T	0.07–0.09	1
Nickel	electrolytic	22	T	0.04	4
Nickel	electrolytic	38	T	0.06	4
Nickel	electrolytic	260	T	0.07	4
Nickel	electrolytic	538	T	0.10	4
Nickel	electroplated, polished	20	T	0.05	2
Nickel	electroplated on iron, polished	22	T	0.045	4
Nickel	electroplated on iron, unpolished	20	T	0.11–0.40	1

1	2	3	4	5	6
Nickel	electroplated on iron, unpolished	22	T	0.11	4
Nickel	oxidized	200	T	0.37	2
Nickel	oxidized	227	T	0.37	4
Nickel	oxidized	1227	T	0.85	4
Nickel	oxidized at 600°C	200–600	T	0.37–0.48	1
Nickel	polished	122	T	0.045	4
Nickel	wire	200–1000	T	0.1–0.2	1
Nickel oxide		500–650	T	0.52–0.59	1
Nickel oxide		1000–1250	T	0.75–0.86	1
Oil, lubricating	0.025 mm film	20	T	0.27	2
Oil, lubricating	0.050 mm film	20	T	0.46	2
Oil, lubricating	0.125 mm film	20	T	0.72	2
Oil, lubricating	film on Ni base: Ni base only	20	T	0.05	2
Oil, lubricating	thick coating	20	T	0.82	2
Paint	8 different colors and qualities	70	LW	0.92–0.94	9
Paint	8 different colors and qualities	70	SW	0.88–0.96	9
Paint	Aluminum, various ages	50–100	T	0.27–0.67	1
Paint	cadmium yellow		T	0.28–0.33	1
Paint	chrome green		T	0.65–0.70	1
Paint	cobalt blue		T	0.7–0.8	1
Paint	oil	17	SW	0.87	5
Paint	oil, black flat	20	SW	0.94	6
Paint	oil, black gloss	20	SW	0.92	6
Paint	oil, gray flat	20	SW	0.97	6
Paint	oil, gray gloss	20	SW	0.96	6
Paint	oil, various colors	100	T	0.92–0.96	1

19

1	2	3	4	5	6
Paint	oil based, average of 16 colors	100	T	0.94	2
Paint	plastic, black	20	SW	0.95	6
Paint	plastic, white	20	SW	0.84	6
Paper	4 different colors	70	LW	0.92–0.94	9
Paper	4 different colors	70	SW	0.68–0.74	9
Paper	black		T	0.90	1
Paper	black, dull		T	0.94	1
Paper	black, dull	70	LW	0.89	9
Paper	black, dull	70	SW	0.86	9
Paper	blue, dark		T	0.84	1
Paper	coated with black lacquer		T	0.93	1
Paper	green		T	0.85	1
Paper	red		T	0.76	1
Paper	white	20	T	0.7–0.9	1
Paper	white, 3 different glosses	70	LW	0.88–0.90	9
Paper	white, 3 different glosses	70	SW	0.76–0.78	9
Paper	white bond	20	T	0.93	2
Paper	yellow		T	0.72	1
Plaster		17	SW	0.86	5
Plaster	plasterboard, untreated	20	SW	0.90	6
Plaster	rough coat	20	T	0.91	2
Plastic	glass fibre laminate (printed circ. board)	70	LW	0.91	9
Plastic	glass fibre laminate (printed circ. board)	70	SW	0.94	9

1	2	3	4	5	6
Plastic	polyurethane isolation board	70	LW	0.55	9
Plastic	polyurethane isolation board	70	SW	0.29	9
Plastic	PVC, plastic floor, dull, structured	70	LW	0.93	9
Plastic	PVC, plastic floor, dull, structured	70	SW	0.94	9
Platinum		17	T	0.016	4
Platinum		22	T	0.03	4
Platinum		100	T	0.05	4
Platinum		260	T	0.06	4
Platinum		538	T	0.10	4
Platinum		1000–1500	T	0.14–0.18	1
Platinum		1094	T	0.18	4
Platinum	pure, polished	200–600	T	0.05–0.10	1
Platinum	ribbon	900–1100	T	0.12–0.17	1
Platinum	wire	50–200	T	0.06–0.07	1
Platinum	wire	500–1000	T	0.10–0.16	1
Platinum	wire	1400	T	0.18	1
Porcelain	glazed	20	T	0.92	1
Porcelain	white, shiny		T	0.70–0.75	1
Rubber	hard	20	T	0.95	1
Rubber	soft, gray, rough	20	T	0.95	1
Sand			T	0.60	1
Sand		20	T	0.90	2
Sandstone	polished	19	LLW	0.909	8
Sandstone	rough	19	LLW	0.935	8
Silver	polished	100	T	0.03	2
Silver	pure, polished	200–600	T	0.02–0.03	1

1	2	3	4	5	6
Skin	human	32	T	0.98	2
Slag	boiler	0–100	T	0.97–0.93	1
Slag	boiler	200–500	T	0.89–0.78	1
Slag	boiler	600–1200	T	0.76–0.70	1
Slag	boiler	1400–1800	T	0.69–0.67	1
Snow: See Water					
Soil	dry	20	T	0.92	2
Soil	saturated with water	20	T	0.95	2
Stainless steel	alloy, 8% Ni, 18% Cr	500	T	0.35	1
Stainless steel	rolled	700	T	0.45	1
Stainless steel	sandblasted	700	T	0.70	1
Stainless steel	sheet, polished	70	LW	0.14	9
Stainless steel	sheet, polished	70	SW	0.18	9
Stainless steel	sheet, untreated, somewhat scratched	70	LW	0.28	9
Stainless steel	sheet, untreated, somewhat scratched	70	SW	0.30	9
Stainless steel	type 18-8, buffed	20	T	0.16	2
Stainless steel	type 18-8, oxidized at 800°C	60	T	0.85	2
Stucco	rough, lime	10–90	T	0.91	1
Styrofoam	insulation	37	SW	0.60	7
Tar			T	0.79–0.84	1
Tar	paper	20	T	0.91–0.93	1
Tile	glazed	17	SW	0.94	5
Tin	burnished	20–50	T	0.04–0.06	1
Tin	tin–plated sheet iron	100	T	0.07	2

1	2	3	4	5	6
Titanium	oxidized at 540°C	200	T	0.40	1
Titanium	oxidized at 540°C	500	T	0.50	1
Titanium	oxidized at 540°C	1000	T	0.60	1
Titanium	polished	200	T	0.15	1
Titanium	polished	500	T	0.20	1
Titanium	polished	1000	T	0.36	1
Tungsten		200	T	0.05	1
Tungsten		600–1000	T	0.1–0.16	1
Tungsten		1500–2200	T	0.24–0.31	1
Tungsten	filament	3300	T	0.39	1
Varnish	flat	20	SW	0.93	6
Varnish	on oak parquet floor	70	LW	0.90–0.93	9
Varnish	on oak parquet floor	70	SW	0.90	9
Wallpaper	slight pattern, light gray	20	SW	0.85	6
Wallpaper	slight pattern, red	20	SW	0.90	6
Water	distilled	20	T	0.96	2
Water	frost crystals	–10	T	0.98	2
Water	ice, covered with heavy frost	0	T	0.98	1
Water	ice, smooth	–10	T	0.96	2
Water	ice, smooth	0	T	0.97	1
Water	layer >0.1 mm thick	0–100	T	0.95–0.98	1
Water	snow		T	0.8	1
Water	snow	–10	T	0.85	2
Wood		17	SW	0.98	5
Wood		19	LLW	0.962	8
Wood	ground		T	0.5–0.7	1

1	**2**	**3**	**4**	**5**	**6**
Wood	pine, 4 different samples	70	LW	0.81–0.89	9
Wood	pine, 4 different samples	70	SW	0.67–0.75	9
Wood	planed	20	T	0.8–0.9	1
Wood	planed oak	20	T	0.90	2
Wood	planed oak	70	LW	0.88	9
Wood	planed oak	70	SW	0.77	9
Wood	plywood, smooth, dry	36	SW	0.82	7
Wood	plywood, untreat-ed	20	SW	0.83	6
Wood	white, damp	20	T	0.7–0.8	1
Zinc	oxidized at 400°C	400	T	0.11	1
Zinc	oxidized surface	1000–1200	T	0.50–0.60	1
Zinc	polished	200–300	T	0.04–0.05	1
Zinc	sheet	50	T	0.20	1

INTENTIONALLY LEFT BLANK

A note on the technical production of this publication

This publication was produced using XML—the *eXtensible Markup Language*. For more information about XML, please visit http://www.w3.org/XML/

A note on the typeface used in this publication

This publication was typeset using Swiss 721, which is Bitstream's pan-European version of the Helvetica™ typeface. Helvetica™ was designed by Max Miedinger (1910–1980).

List of effective files

20235103.xml a24
20235203.xml a21
20235303.xml a19
20236703.xml a57
20238503.xml a9
20238703.xml b8
20241103.xml a14
20248903.xml a8
20249003.xml a11
20249103.xml a4
20249403.xml a6
20249503.xml a5
20249603.xml a5
20249703.xml a6
20249803.xml a2
20249903.xml a2
20250403.xml a21
20250503.xml a4
20251003.xml a2
20254903.xml a75a2
20255303.xml a12
20255603.xml a10
20257003.xml a40
20287303.xml a9
R0034.rcp a7
config.xml a5

Corporate Headquarters
FLIR Systems, Inc.
27700 SW Parkway Avenue
Wilsonville, OR 97070
USA
Telephone: +1-800-727-3547
Website: http://www.flir.com